2

NCS TIB 87-19

NATIONAL COMMUNICATIONS SYSTEM

TECHNICAL INFORMATION BULLETIN 87-19

MODIFICATION OF THE GROUP 4 FACSIMILE VALIDATION SYSTEM

MAY 1987

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

ADA197712

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1a REPORT SECURITY CLASSIFICATION
Unclassified

1b RESTRICTIVE MARKINGS

2a SECURITY CLASSIFICATION AUTHORITY

2b DECLASSIFICATION/DOWNGRADING SCHEDULE

3 DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release; distribution unlimited

4 PERFORMING ORGANIZATION REPORT NUMBER(S)
NCS TIB 87-19

5 MONITORING ORGANIZATION REPORT NUMBER(S)

6a NAME OF PERFORMING ORGANIZATION
Delta Information Systems

6b OFFICE SYMBOL (If applicable)

7a NAME OF MONITORING ORGANIZATION

6c ADDRESS (City, State, and ZIP Code)
Horsham Business Center, Bldg. 3
300 Welsh Road
Horsham, PA 19044

7b ADDRESS (City, State, and ZIP Code)

8a NAME OF FUNDING/SPONSORING ORGANIZATION
National Communications System

8b OFFICE SYMBOL (If applicable)
NCS-TS

9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
DCA100-83-C-0047

8c ADDRESS (City, State, and ZIP Code)
Office of Technology & Standards
Washington, D. C. 20305-2010

10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO
P00009 33127K	MD P00009	3	

11 TITLE (Include Security Classification)
Modification of the Group 4 Facsimile Validation System

12 PERSONAL AUTHOR(S)

13a TYPE OF REPORT
Final

13b TIME COVERED
FROM ________ TO ________

14 DATE OF REPORT (Year, Month, Day)
May 1987

15 PAGE COUNT
93

16 SUPPLEMENTARY NOTATION

17 COSATI CODES

FIELD	GROUP	SUB-GROUP

18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Facsimile
Packet Switched Data Network (PSDN)
Group 4
Public Switched Telephone Network (PSTN)

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

The purpose of this task was to update the Group 4 Validation System in accordance with the latest CCITT Recommendations pertaining to Group 4 Facsimile Equipment. The three sections give a system overview, the system modifications required and validation system testing ~~are presented.~~ One section reviews the overall validation system requirements and the system structure used to meet these requirements. Another section detials the system modifications necessary to conform to the T series recommendations and the layers within the Open Systems Interconnection (OSI) that were effected. The last section discusses Group 4 validation system testing and any operational features changed by the implemented changes.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS

21 ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a NAME OF RESPONSIBLE INDIVIDUAL
J. Orndorff

22b TELEPHONE (Include Area Code)
202-692-2124

22c OFFICE SYMBOL
NCS-TS

NCS TECHNICAL INFORMATION BULLETIN 87-19

MODIFICATION OF THE GROUP 4 FACSIMILE VALIDATION SYSTEM

PROJECT OFFICER	APPROVED FOR PUBLICATION:
	Dennis Bodson
DENNIS BODSON Senior Electronics Engineer Office of NCS Technology and Standards	DENNIS BODSON Assistant Manager Office of NCS Technology and Standards

FOREWORD

Among the responsibilities assigned to the Office of the Manager, National Communications System, is the management of the Federal Telecommunication Standards Program. Under this program, the NCS, with the assistance of the Federal Telecommunication Standards Committee identifies, develops, and coordinates proposed Federal Standards which either contribute to the interoperability of functionally similar Federal telecommunication systems or to the achievement of a compatible and efficient interface between computer and telecommunication systems. In developing and coordinating these standards, a considerable amount of effort is expended in initiating and pursuing joint standards development efforts with appropriate technical committees of the Electronics Industries Association, the American National Standards Institute, the International Organization for Standardization, and the International Telegraph and Telephone Consultative Committee of the International Telecommunication Union. This Technical Information Bulletin presents an overview of an effort which is contributing to the development of compatible Federal, national, and international standards in the area of facsimile. It has been prepared to inform interested Federal activities of the progress of these efforts. Any comments, inputs or statements of requirements which could assist in the advancement of this work are welcome and should be addressed to:

Office of the Manager
National Communications System
ATTN: NCS-TS
Washington, DC 20305-2010

MODIFICATION OF THE GROUP 4 FACSIMILE VALIDATION SYSTEM

May, 1987

Final Report

Submitted to:

NATIONAL COMMUNICATIONS SYSTEM
Office of Technology and Standards
Washington, DC 20305

Contracting Agency:

DEFENSE COMMUNICATIONS AGENCY
Contract Number DCA100-83-C-0047
MODIFICATION P00009 - TASK 3

DELTA INFORMATION SYSTEMS, INC.
Horsham Business Center, Bldg. 3
300 Welsh Road
Horsham PA 19044
TEL: (215) 657-5270 FAX: (215) 657-5273

Table of Contents

1.0 Introduction

This document summarizes work performed by Delta Information Systems, Inc., for the Office of Technology and Standards of the National Communications System, an organization of the U. S. Government, headed by National Communications System Assistant Manager of Technology and Standards, Dennis Bodson. Mr. Bodson is responsible for the management of the Federal Telecommunications Standards Program, which develops telecommunication standards, the use of which is mandatory for all Federal agencies. Developed under a previous contract for the National Communication System was a software system to validate Group 4 Facsimile Equipment when operating over a public switched telephone network (PSTN) or a packet switched data network (PSDN). The validation system was developed based on the then currently existing CCITT S Series Recommendations. Subsequent to the development of the validation system the CCITT S Series Recommendations pertaining to the Telematic Services were updated and renamed as T Series Recommendations. (e.g S.70 to T.70 and S.a to T.73). The purpose of this task, performed under contract number DCA100-83-C-0047, was to update the Group 4 Validation System in accordance with the latest CCITT Recommendations pertaining to Group 4 Facsimile Equipment.

In the following three sections a system overview, the system modifications required and Validation system testing are presented. Section 2.0 reviews the overall Validation system requirements and the system structure used to meet these requirements. Section 3.0 details the system modifications necessary to conform to the T Series Recommendations and the layers within the Open Systems Interconnection (OSI) that were effected. Section 4.0 discusses Group 4 Validation System testing and any operational features changed by the implemented changes.

2.0 Group 4 Validation System Overview

The primary purpose of the Group 4 Validation system is the testing/evaluation of a Group 4 Facsimile terminal. Figure 2.1 is a simplified block diagram of the validation system architecture. The function of the validation system is to insure that the Group 4 terminal unit under test (UUT) has properly implemented the protocols required for layers 3 through 6 of the telematic Protocol structure for Group 4 Facsimile equipment and also conforms to the allowable parameter variations (e.g. buffer sizes, timeout periods, etc.) within each protocol layer. Since layer 7, the Application layer, currently has no defined protocol this layer cannot be validated. In addition protocol testing of layers 1 & 2 (Physical layer & Link layer) is not performed. However, any protocol violations or nonrecoverable errors are reported to the higher layers.

Figure 2.2 is a diagram of the Group 4 Validation system structure. The validation system was implemented with layers 3 through 7 and the necessary control routines (e.g. test control, error logging, etc.) using Delta Information System's HP 1000 processor. As seen in Figure 2.2, layers 1 & 2 were implemented using a microprocessor controlled Packet Data Interface (PDI) board developed by Delta Information Systems. The PDI interfaces with the HP 1000 bus and to the appropriate modem for either a Public Switched Telephone Network (PSTN) or a Packet Switched Public Data Network (PSPDN).

The Group 4 Validation System software consists of the

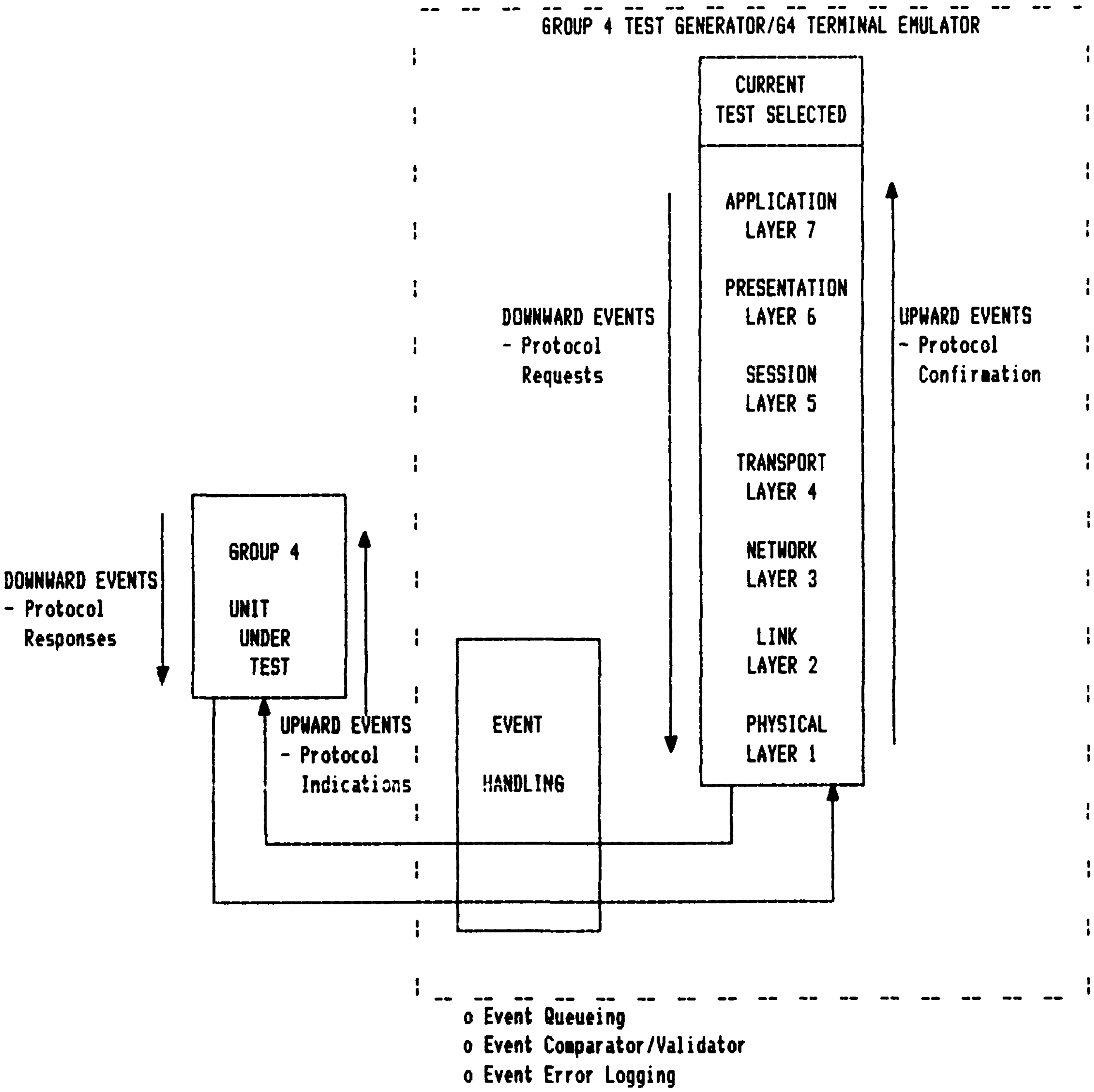

GROUP 4 TEST GENERATOR/G4 TERMINAL EMULATOR
CURRENT TEST SELECTED
APPLICATION LAYER 7
PRESENTATION LAYER 6
SESSION LAYER 5
TRANSPORT LAYER 4
NETWORK LAYER 3
LINK LAYER 2
PHYSICAL LAYER 1
DOWNWARD EVENTS - Protocol Requests
UPWARD EVENTS - Protocol Confirmation
DOWNWARD EVENTS - Protocol Responses
GROUP 4 UNIT UNDER TEST
UPWARD EVENTS - Protocol Indications
EVENT HANDLING
o Event Queueing
o Event Comparator/Validator
o Event Error Logging

FIGURE 2.1 GROUP 4 VALIDATION SYSTEM

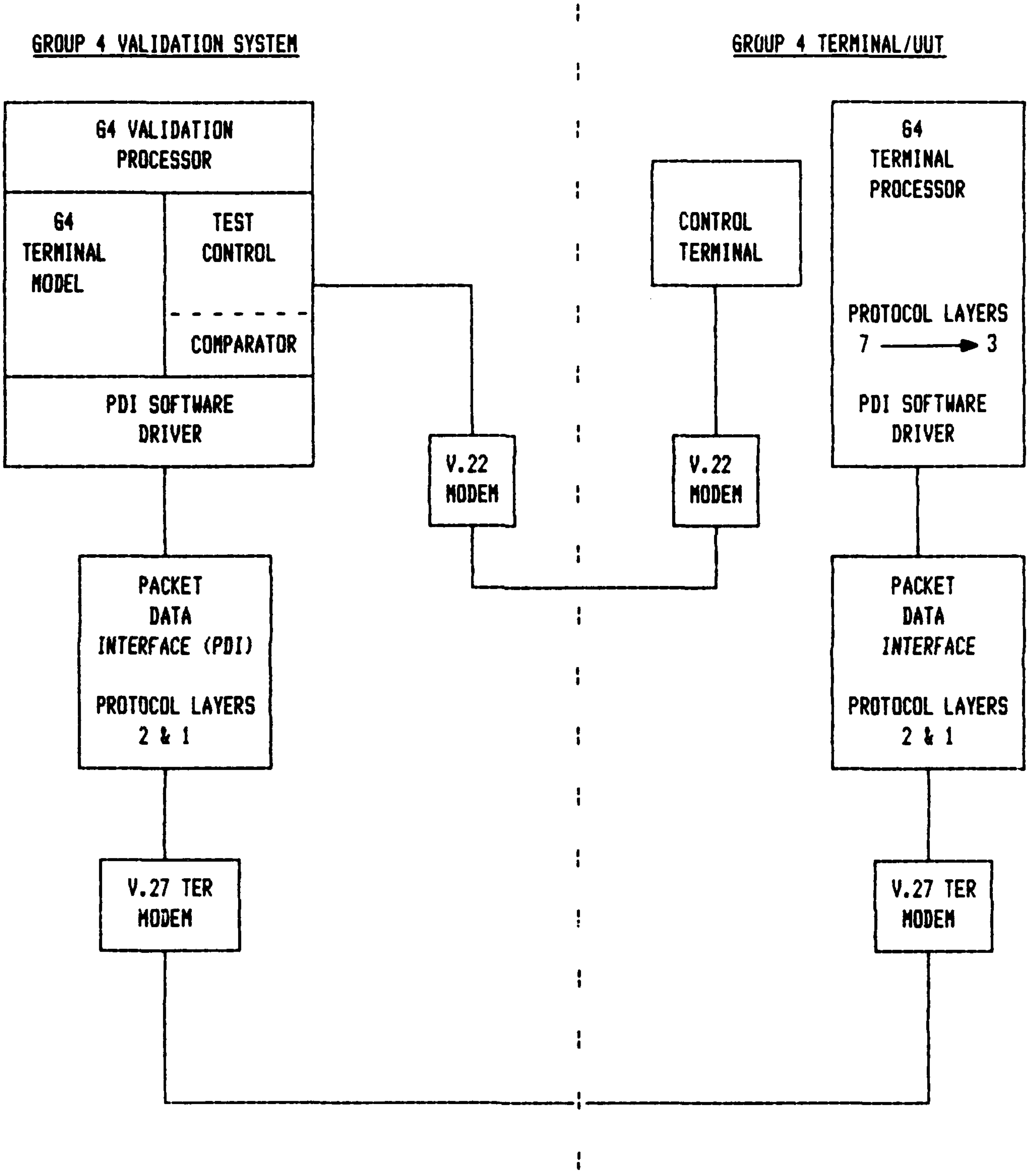

GROUP 4 VALIDATION SYSTEM
GROUP 4 TERMINAL/UUT
G4 VALIDATION PROCESSOR
G4 TERMINAL MODEL
TEST CONTROL
COMPARATOR
PDI SOFTWARE DRIVER
V.22 MODEM
CONTROL TERMINAL
V.22 MODEM
G4 TERMINAL PROCESSOR
PROTOCOL LAYERS
7 → 3
PDI SOFTWARE DRIVER
PACKET DATA INTERFACE (PDI)
PROTOCOL LAYERS 2 & 1
PACKET DATA INTERFACE
PROTOCOL LAYERS 2 & 1
V.27 TER MODEM
V.27 TER MODEM

FIGURE 2.2

system executive or test control side, the Group 4 terminal emulator and a Group 4 Unit Under Test (UUT) side(s). The system executive controls the execution of the selected test(s) and verifies that the Group 4 UUT is correctly executing and responding to the test. The emulator simulates a "model" Group 4 terminal and performs all operations as requested by the validator. A system comparator compares the results of the model and the UUT insuring correct operation. The validation system software as implemented can support both the validator and emulator/UUT sides within the same processor for testing or two separate processors to demonstrate the actual system operation. The software for the validation system was written in Fortran 77 to maximize its transportability between the HP 1000 and a DEC PDP 11 for full system testing. The software design and development was done using top-down structured techniques. This design approach yields software which is highly modular and easy to modify. This modularity will also facilitate changes to the software required by the evolving CCITT Recommendations.

The Group 4 Validation System hardware consists of the Packet Data Interface board, the V.27TER modem and the PSTN/PSPDN (Public Switched Telephone Network/Packet Switched Public Data Newtork) network connection. In addition the processor which is functioning as the validator has a 1200 bps dial-in control port in order to allow the user to communicate with the validation test set. The Packet Data Interface (PDI) board is a DIS-developed assembly designed specifically to support both the

Group 4 validator and emulator functions. The design uses identical hardware to interface to both processors (currently a HP1000 and DEC PDP 11). Module operation is controlled by an imbedded microprocessor executing code developed in a high level language ("C) which permits modularity and flexibility.

The PDI interfaces to the computer by parallel program controlled transfer interfaces. In the case of the HP 1000, this interface was also developed by DIS to permit the same interconnect to be presented to the PDI as that seen from the DEC PDP 11.

3.0 Validation System Modifications

The Group 4 Validation System was developed in accordance with the Open Systems Interconnection (OSI) 7 Layer Architecture. Since its initial implementation, the CCITT S Series Recommendations pertaining to the Telematic Services have been updated and renamed as T Series Recommendations. In addition a first release of CCITT Recommendation T.73 governing the Presentation Layer services has been completed. T.73 defines the Document Interface Protocol for the telematic services and with its inception has moved the document structure parameters from the Session Layer (Layer 5) to Layer 6.

Recommendation S.62 which defined the Session Layer services for Teletex was updated to include the control procedures for both Teletex and Group 4 Facsimile and was reissued as T.62. As was mentioned above, the document structure parameters were removed from the Session layer and a Session User Data (SUD) parameter id was added to facilitate the transfer of the Presentation Layer parameters. In addition T.62 was modified to insure compatibility with Recommendations X.215 and X.225 which specify the OSI session services and protocol.

Recommendation S.70 which defined the network independent basic transport for Teletex was updated to include both Teletex and Group 4 Facsimile and was reissued as T.70. As with T.62, T.70 was also modified to insure compatibility with Recommendations X.214 and X.224 which specify the OSI transport services and protocol. Shown in Figure 3.1 are the relationships between the T Series recommendations and the OSI protocols.

3.1.1 Presentation Layer - T.73

T.73 defines the document interchange protocol to be used above session services within the Telematic service when a document structure is required for mixed mode and Group 4 facsimile. The interchange representation of a document consists of a sequence of protocol elements. Four types of such elements have been defined: presentation capabilities descriptor, document profile description, layout descriptors and text units.

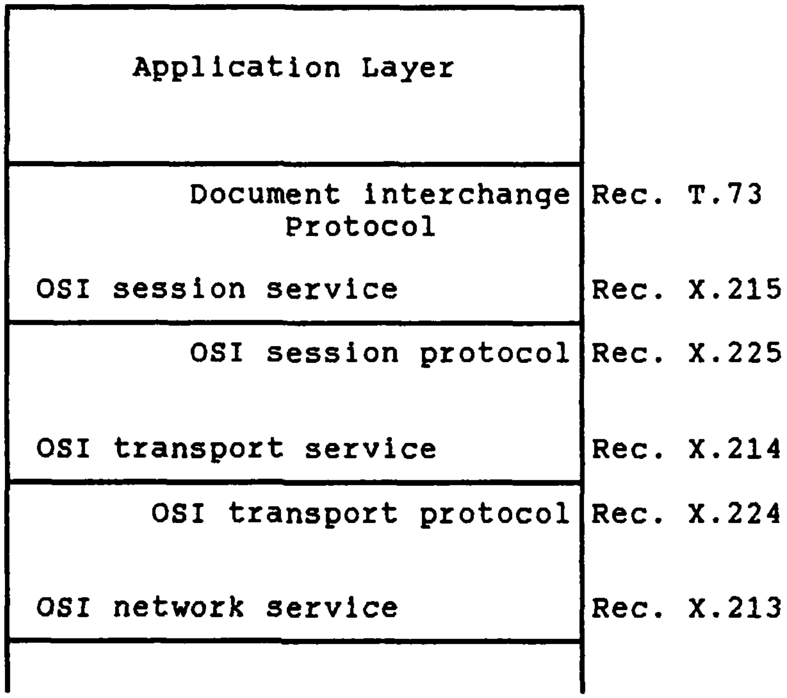
The presentation capabilities descriptor consists of the following four parts:

- basic terminal characteristics
- interchange format
- non-basic terminal capabilities
- non-basic structural capabilities

The document profile descriptor consists of the following four parts:

- generic layout structure
- specific layout structure
- present capabilities to be provided
- document profile attributes

The layout descriptor is a data element consisting of subordinate data elements and elementary data items. The elementary data



Relationships between T Series Recommendations and OSI protocols

Figure 3.1

items are a small number of basic data types such as numbers, character strings and bit strings. The text unit is a data element consisting of two parts:

- attributes of document content
- data elements representing a portion of document

In order for the presentation layer to communicate the protocol elements it uses the following session service primitives:

Session Connection

- S-CONNECT request, S-CONNECT indication
- S-CONNECT response, S-CONNECT confirm

Session Capabilities Exchange

- S-CAPAB DATA request, S-CAPAB DATA indication
- S-CAPAB DATA response, S-CAPAB DATA confirm

Session Activity Begin

- S-ACT BEG request, S-ACT BEG indication

Session Normal Data

- S-DATA request, S-DATA indication

The presentation capabilities descriptor protocol element is transfered on the session connection and session capabilities exchange. In addition the presentation capabilities descriptor can be sent on the session activity begin. The following protocol elements are sent on the session normal data primitive:

- Document profile description
- Layout descriptors
- Text unit

A number of possible document types are defined in T.73. A terminal may negotiate the capability to use several types of documents within a session using the Session Connect or Session Capabilities Exchange primitives, however only one type of document can be sent during the document transfer.

Since the T.73 Recommendation was incomplete at the time of its release DIS added to the Primitives specified in T.73 to allow for page boundary checkpointing, session termination and a session abort. The following session service primitives were used:

Session Page Checkpoint

- S-Sync-Minor request, S-Sync-Minor Indication
- S-Sync-Minor respomse, S-Sync-Minor confirm

Session Disconnect

- S-Release request, S-Release indication
- S-Release response, S-Release confirm

Session User Abort

- S-U-ABT request, S-U-ABT indication
- S-U-ABT response, S-U-ABT confirm

3.1.2 Session Layer - T.62

T.62 Recommendation defines the end-to-end procedures to be used within the Telex and Group 4 facsimile services. A full set of session service primitives was defined in Annex H of T.62 along with the state diagrams governing their use. T.62 also specified the session protocol data units (SPDUs) associated with each of the session layer service primitives. Listed below are the session service primitives now available to the presentation layer and the SPDUs for each.

Session Connection

S-CON Request, S-CON Indication - CSS
S-CON Response, S-CON Confirm - RSSP or RSSN

Session Release

S-REL Request, S-REL Indication - CSE
S-REL Response, S-REL Confirm - RSEP

Session User Abort

S-U-ABT Request, S-U-ABT Indication - CSA
S-U-ABT Response, S-U-ABT Confirm - RSAP

Session Provider Abort

S-P-ABT Indication

Session Control Give

S-CTRL-GIVE Request, S-CTRL-GIVE Indication - CSCC
S-CTRL-GIVE Response, S-CTRL-GIVE Confirm - RSCCP

Session Token Please

S-TOKEN-PLS Request, S-TOKEN-PLS Indication - RSUI

Session Activity Begin

S-ACT-BEG Request, S-ACT-BEG Indication - CSUI/CDS
CSUI/CDC

Session Data Transfer

S-DATA Request, S-DATA Indication CSUI/CDUI

Session Synchronization Minor

S-SYNC-MIN Request, S-SYNC-MIN Indication - CSUI/CDPB
S-SYNC-MIN Response, S-SYNC-MIN Confirm - RSUI/RDPBP
or
S-U-EXPT Request, S-U-EXPT Indication - RSUI/RDPBN

Session Activity End

S-ACT-END Request, S-ACT-END Indication - CSUI/CDE
S-ACT-END Response, S-ACT-END Confirm - RSUI/RDEP
or
S-U-EXPT Request, S-U-EXPT Indication - RSUI/RDPBN

Session Activity Interrupt

S-ACT-INT Request, S-ACT-INT Indication - CSUI/CDR
S-ACT-INT Response, S-ACT-INT Confirm - RSUI/RDRP

Session Activity Discard

S-ACT-DCAD Request, S-ACT-DCAD Indication - CSUI/CDD
S-ACT-DCAD Response, S-ACT-DCAD Confirm - RSUI/RDDP

Session Capability Data

S-CAPAB-DATA Request, S-CAPAB-DATA Indication - CDCL
S-CAPAB-DATA Response, S-CAPAB-DATA Confirm - RDCLP

Session User Execption Reporting

S-U-EXPT Request, S-U-EXPT Indication - RSUI/RDPBN

Session Provider Execption Reporting

S-P-EXPT Indication

Although the session layer software is a full implementation of T.62, only those session primitives listed in Section 3.1.1 are currently being used by the presentation layer. It also reflects as closely as possible the session and document state transition diagrams as presented in Annex H of T.62 Recommendation.

In addition to the formalization of the protocol service primitives, it was necessary to delete from the T.62 Parameter Group Identifiers (PGIs) and Parameter Identifiers (PIs) those document related parameters that were moved to the Presentation layer. Added to the PGIs in the session layer was PGI for Session User Data (hex C1) for the transfer of presentation layer data.

3.1.3 Transport Layer - T.70

Recommendation T.70 specifies the network independent basic transport for Teletex and Group 4 Facsimile services. As with T.62, a full set of transport service primitives was defined in Annex A of T.70 along with the state diagrams governing their use. T.70 also specified the transport protocol data units (TPDUs) associated with each of the transport layer service primitives. Listed below are the tranport service primitives now available to the session layer and the TPDUs for each.

Tranport Connection

T-CONNECT Request, T-CONNECT Indication - TCR
T-CONNECT Response, T-CONNECT Confirm - TCA or TCC

Transport Release

T-DISCONNECT Request, T-DISCONNECT Indication
T-DISCONNECT Response, T-DISCONNECT Confirm

Transport Transfer Phase

T-DATA Request, T-DATA Indication - TDT

Transport Error Reporting

T-EXCEPTION Indication

In addition changes were made within the transport service to allow for transport data block size negotiations, extended addressing for multiterminal configurations and the handling of procedural errors.

4.0 Validation System Testing

4.1 Validation System Operation Overview

The Group 4 Validation System consists of the system executive or test control side, the Group 4 terminal emulator and a Group 4 Unit Under Test (UUT) side(s). The system executive controls the execution of the selected test(s) and verifies that the Group 4 UUT is correctly executing and responding to the test. The emulator simulates a "model" Group 4 Terminal and performs all operation as requested by the validator. A system comparator compares the results of the model and the UUT insuring correct operation. The validation system software as implemented can support both the validator and emulator/UUT sides within the same processor for testing, or two separate processors to demonstrate the final system configuration.

The performance of a test by the validator can be roughly divided into five phases:

- Test Selection
- Test Validation
- Test Configuration
- Test Execution
- Test Post-mortems

The Test Selection phase involves the determination of the specific test to be performed and any "special" parameters involved. It involves the processing of a "test select" file which names the test to be used, and which supplies the parameters unique to this particular performance.

Test validation involves verifying that the test file named, if "new", is in the proper form, and processing its (default) parameters for later merging with those specified by the test select file. If the "repeat" (TEST=*) or "default" (TEST=&) options are chosen, no validation is necessary escept confirming the existence of a previously configured test.

Test Configuration is the most complexx of the three "test setup" phases. First, the "repeat" (*) and "default" (&) parameter "values" specified in the test select file are "filled in" from the last-executed test performance or the "as read in" test parameters as appropriate. The test select parameters are then merged with either those from the last performance (if test reapeat is specified) or those of the "standard" (as read in) test. Specific test parameters are then processed, first the "global" ones which must be first processed, then those "local" to specific layers, sides, etc. The layer-specific "local" parameters include configuration indications which specify how each layer and half-layer is implemented (e.g., accessible as software, inaccessible in hardware, an interface between the two, or just "not there"); the next stage in the configuration phase validates that the layer to layer configurations are consistent. Finally, any "leftover" parameters are processed and reported on.

Test Execution involes initialization of global and layer-specific status to "startup" condition, preparation of the test "script" (run data) for processing, and transfer of control to the event-driven test execution. Global initialization involves setting the event queue "empty" (ready to receive events), and resetting test-level status indicator. Initialization of the various layer/side/end combinations depends on how each is configured.

Following the successful or unsuccessful performance of a test, its results are reported. This involves printouts of both completed (fully processed) events, and events whose processing was not completed, either because of lack of time or error conditions raised. This phase also serves to prepare the system for the next test performance.

4.2 Validation System Integration Testing

Since the changes defined by the T-Series recommendations only effected the Presentation, Session and Transport layers of the validation system no new testing was required for the Network, Link and Physical layers. In addition because of the top-down design methodology used in developing the validation system it was not necessary to implement and test all the changes at one time. Starting with the presentation layer and working downward the changes for each layer were implemented and tested by having each layer communicate with its peer layer without requiring the layers below them. This was done by using half-layer modules (e.g. layer 5.5) which communicated with each other, 5.5 to 5.5, rather than the layer below them, 5.5 to 5. This minimized the integration time since only a single module/layer was being changed and tested at a time. Once the integration testing of the module(s) was completed, the half-layer module was restored to the original and testing of the next layer down was begun. This procedure continued until all the new modules for the changed layers had been integrated and tested within the validation system.

4.3 Validation System System Testing

With completion of the integration testing, the currently existing Test Selection and Test Data files were modified to reflect the changes made to the Presentation, Session and Transport layers. This consisted primarily of updating the test run data files to use the Session layer and Transport layer service primitives as defined in the T-Series recommendations. They were also updated to include any new layer specific variables/parameters as specified in the recommendations. The full validation system was then run, exercising the full end-to-end capabilities of a Group 4 Facsimile Terminal.

5.0 Recommendations

The Group 4 Validation System was developed in accordance with the Open Systems Interconnection (OSI) Layer architecture (See Figure 5.1). Over the past year considerable work has been done within CCITT Study Group VIII pertaining to Group 4 Facsimile. This work focused primarily on the Application, Presentation and Session layers. In addition a preliminary investigation into the use of Integrated Services Digital Network (ISDN) within the Transport layer was begun. Shown in Figure 5.2 are the current Recommendations for Group 4 Facsimile. The following is an overview of the current activity within the CCITT.

As is shown in Figure 5.1, there was no CCITT Recommendation defining the Application layer for Group 4 Facsimile when the validation system was originally implemented. Because of this the layer 7 software in the Validation system was designed using as a base the OSI Application Layer Model and its associated Common Application Service Elements (CASE). One major area of work by Study Group VIII was in the establishment of the T.400 series of Recommendations defining the Application layer. The T.400 Recommendations define a Document Transfer and Manipulation (DTAM) Service, a Document Architecture handled by DTAM and a DTAM Protocol available within the Application layer of the reference model. The DTAM defined in this series of recommendations is one of the Application Service Elements (ASE), which is specifically designed for document handling. It is

CCITT RECOMMENDATIONS BY ISO LAYER

FOR GROUP 4 FACSIMILE

<table>
<tr><td colspan="2">APPLICATION LAYER
NO CURRENT RECOMMENDATION</td><td>LAYER 7</td></tr>
<tr><td colspan="2">PRESENTATION LAYER
CCITT T.73</td><td>LAYER 6</td></tr>
<tr><td colspan="2">SESSION LAYER
CCITT T.62</td><td>LAYER 5</td></tr>
<tr><td colspan="2">TRANSPORT LAYER
CCITT T.70</td><td>LAYER 4</td></tr>
<tr><td colspan="2">NETWORK LAYER
CCITT X.25</td><td>LAYER 3</td></tr>
<tr><td colspan="2">LINK LAYER
CCITT X.25</td><td>LAYER 2</td></tr>
<tr><td>PHYSICAL
X.24
PSTN</td><td>LAYER
X.21
PSPDN</td><td>LAYER 1</td></tr>
</table>

Figure 5.1

CCITT RECOMMENDATIONS BY ISO LAYER

FOR GROUP 4 FACSIMILE

APPLICATION LAYER		LAYER 7
CCITT T.400 SERIES		
PRESENTATION LAYER		LAYER 6
SESSION LAYER CCITT T.62 UNDER REVIEW TO BE REPLACED BY T.62 bis		LAYER 5
TRANSPORT LAYER CCITT T.70		LAYER 4
NETWORK LAYER CCITT X.25		LAYER 3
LINK LAYER CCITT X.25		LAYER 2
PHYSICAL X.24 PSTN	LAYER X.21 PSPDN	LAYER 1

Figure 5.2

concerned with identifiable bodies of information which can be treated as documents, and may be stored within opens systems or transferred and manipulated between application processes. Specifically Recommendations T.4Ya, T.4Yb and T.4Yc define application rules, a basic DTAM service and protocol respectively. They provide sufficient facilities to support DTAM, and establish a framework for DTAM management.

Along with the development of the T.400 series of Recommendations for the application layer, CCITT Recommendation T.73 is being studied and is evolving in the T.400 series. T.73 was developed towards the end of the 1980-1984 CCITT study period and became a full recommendation in 1984. It was based on early concepts which have undergone substantial changes. These changes have been reflected in the new T.400 series. In addition, the presentation layer service as presented to the application layer have been specified in CCITT Recommendation X.216, Presentation Service Definition for Open Systems Interconnection.

The Session Layer control procedures for Teletex and Group 4 Facsimile are descibed in CCITT Recommendation T.62. This recommendation defines the end-to-end procedures which are used within the session layer of the Group 4 Validation system. Currently CCITT Study Group VIII is reviewing T.62 with the intent that a new recommendation, T.62bis, will supercede it. It is the object of this review that X.215, Session Service Definition for OSI, and X.225, Session Protocol Specification for OSI, along with the new recommendation have the same level of detail and accuracy as the current T.62.

CCITT Recommendation T.70 defines the transport layer procedure to be used by equipment connected to CSPDN, PSPDN and PSTN. The transport layer procedures for Integrated Services Digital Networks (ISDN) is currently being studied within Study Group VIII. Since some European countries intend to introduce ISDN within the next few years, there is an urgent need to have a stable CCITT recommendation available as soon as possible. Consideration of Liaison Statements, Protocol Selection for Network Access Layer and Protocol Selection for Application Layer have already been identified for study.

It is recommended that all activity within the CCITT Study Group VIII pertaining to both Group 4 Facsimile and the OSI Reference Model be monitored and that the Validation System be updated to reflect the latest available Recommendations. This is especially important in regard to Application and Presentation Layers where no recommendation or an incomplete one was all that was available for the currently implemented Group 4 Validation System.

APPENDIX A

VALIDATION SYSTEM HARDWARE DESCRIPTION

APPENDIX A

PDI WORK OVERVIEW

GENERAL

The Packet Data Interface (PDI) hardware was designed and two units constructed to provide a communications path between the Group 4 Facsimile Tester and a simulated Unit Under Test. The units implement the HDLC/LAPB physical and link layer of the OSI layered protocol.

HARDWARE DESIGN

The unit was designed to interface to either a Digital Equipment Corporation or Hewlett Packard computer using parallel data transfers. The link interface consists of an X.25 protocol controller I.C. and 4800 baud modem. A 68000 microprocessor provides the data path and control functions for the PDI. The design also includes considerations for operation over both the Public Switched Telephone Network (PSTN) and Packet Switched Public Data Network (PSPDN).

The design of the module was completed and tests which established a link between the two modules with intervening modems were successfully completed. These tests utilized the X.25 protocol and included the following operations:

- Link up
- Data transmission
- Retransmission of corrupted packets
- Disconnect
- Data Reception
- Timeout operations

FIRMWARE DESIGN

Firmware is provided for the 68000 processor in the PDI which controls data flow, link operations, and provides diagnostic maintenance capabilities. The code is written primarily in the "C" language.

HP INTERFACES

In order to provide an identical hardware and software interface to the PDI from both the DEC and HP computers, it was necessary to also design and construct an interface for the HP machine. This interface is physically located within the HP I/O card cage and interfaces to the HP backplane. The output of the card appears identical to the DEC interface with which the PDI is designed to interconnect.

For testing, two of these HP interfaces were constructed and tested to permit testing of the system using HP-to-HP communications. Once constructed, the interface between the HP and the PDI was tested by verifying the transmission and reception of data with the PDI from the HP-driver.

DOCUMENTATION

To adequately document the PDI assembly a manual was written which contains the following information:

Overall Description   Installation Procedures
Operation Descriptions   Diagnostic Testing
Capabilities
Overview Design Description   Detailed Design Description
Drawings

CHAPTER 1

INTRODUCTION

1.1 GENERAL

This manual describes the installation, operation, and design of a Packet Data Interface (PDI). The PDI is used to implement the physical and link layer (HDLC/LAPB) in a system designed to validate Group 4 Facsimile equipment over a Public Switched Telephone Network (PSTN) or Packet Switched Public Data Network (PSPDN). As such it provides a bi-directional data path between a host computer and the Network. The PDI is a self-contained, rack mounted unit. It interfaces to a modem using a 4.8K baud RS-232C link for the network connection. It interfaces to the host computer using a parallel interface compatible with the standard Digital Equipment Corporation computers.

Figure 1-1 presents the front panel layout and Figure 1-2 presents a typical system interconnect diagram using two PDI units in support of Group 4 testing.

1.2 OVERVIEW

There are two key LSI devices in the PDI. The first is a 68000 32 bit processor. This processor controls the flow of data in both directions as well as supporting various protocol dependent func'ions. The 68000 handles the packet transmission between the host computer and the PDI on-board memory. The second key LSI device is a Western Digital WD2511 protocol chip. The WD2511 moves all data between the on-board memory and the modem in the bit-oriented, full duplex serial format which conforms to CCITT X.25 with programmable enhancements. The 68000 processor also controls the ancillary functions of WD2511 setup and modem operations such as dialing and half duplex operations as appropriate to operate on either PSTN or PSPDN.

1.3 PDI CHARACTERISTICS

The PDI is configured as follows:

- MC68000 processor, 32 bit registers, 16 bit data path, 6Mhz operation
- Program storage: 16K words of UV-EPROM
- RAM memory: 128K bytes of dynamic RAM shared for program RAM, processor stack, and input/output packet storage.
- 16 bit bi-directional parallel data interface with host
- Two asynchronous, 9600 baud, serial RS232 ports for test purposes
- WD2511 X.25 packet network interface, providing DMA transfers of data to the on-board RAM
- 1 millisecond timer to facilitate half duplex operations per T.71
- Modem controls for data handling and dialing

1.4 FIRMWARE

The 68000 processor executes a program stored in PROM on the board. This program allows the board to be configured as either the caller or receiver and therefore allows the same board to be used as tester or Unit Under Test model. The firmware operates as directed by commands received from the host computer to establish a link, send data packets, receive data packets, and report status. It also contains diagnostic routines which support maintenance and troubleshooting.

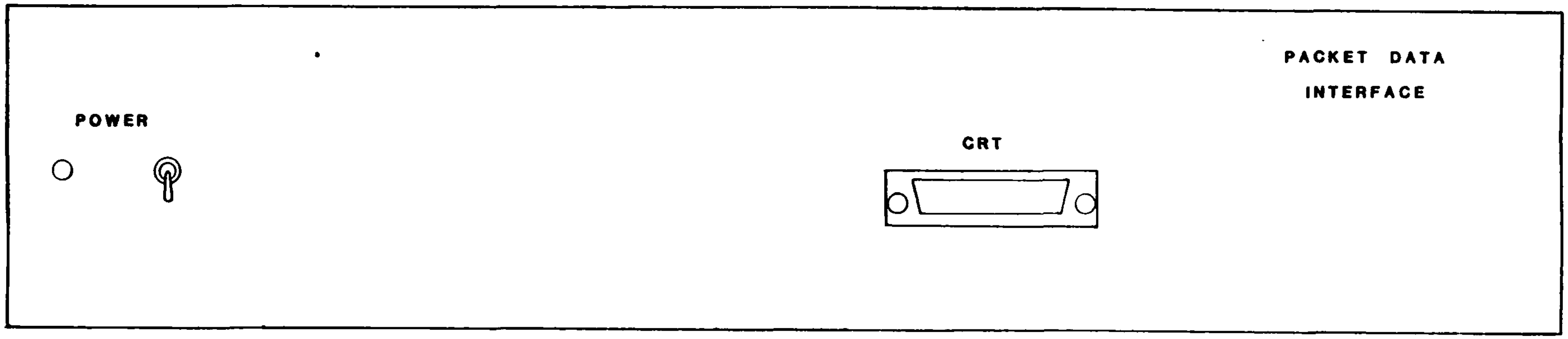


FIGURE 1-1, FRONT PANEL DRAWING

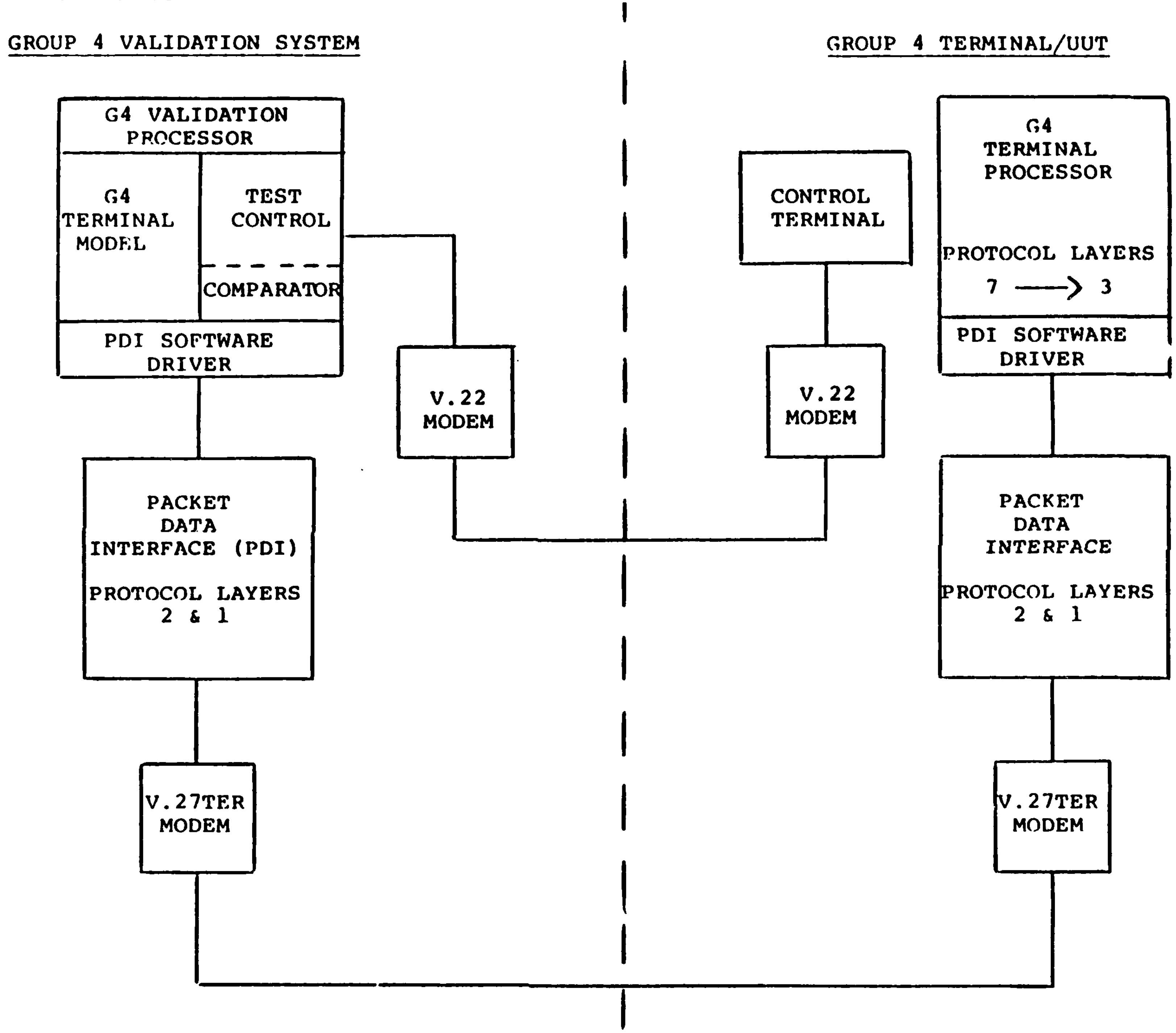


FIGURE 1-2
TYPICAL SYSTEM INTERCONNECT

CHAPTER 2

INSTALLATION

2.1 GENERAL

This section provides the information necessary for site planning, installation, and cabling of the PDI.

2.2 SITE PLANNING

The PDI is supplied in a chassis for mounting in a standard 19" EIA rack. The unit requires a depth of 12" and 3 1/2" of front panel clearance.

The unit should be placed in a position'which results in the minimum parallel cable run between the PDI and the computer.

The unit requires an AC power source of 115 VAC, 47-63 Hz. A power cable (4 feet long) is provided. There are no fans for cooling. Therefore the ambient air temperature must be maintained between 0 and 25 degrees C.

2.3 INSTALLATION

Due to its light weight, the unit is mounted in the chassis simply using four No. 10-32 x 3/4" pan head screws with nylon or plastic washers to secure the front panel in place.

2.4 CABLING

The following tables present the pinouts of the connectors for the Modem, Host Interface, and Test CRT Port.

TABLE 2-1

Modem Connection Pinout

PIN	SIGNAL	DESCRIPTION
1	Chassis	Chassis ground
2	PAKTD	Transmitter data ouput to modem
3	PAKRD	Receiver data input from modem
4	RTS	Request to send to modem
5	CTS	Clear to send from modem
6	DSR	Data Set Ready from modem
7	GND	Signal ground
8	CD	Carrier detect from modem
12	DIALEN	Dial enable to modem
15	PAKTCK	Transmitter clock input
17	PAKRCK	Receiver clock input
20	DTR	Data Terminal Ready to modem
22	RING	Ring indicator from modem
24	PAKST	Transmitter clock output

TABLE 2-2

HOST INTERFACE PINOUT

PIN	SIGNAL	PIN	SIGNAL
1	IOB0	2	IOB8
3	1	4	9
5	2	6	10
7	3	8	11
9		10	
11	4	12	12
13	5	14	13
15	6	16	14
17	IOB7	18	IOB15
19	GND	20	GND
21	GND	22	GND
23	GND	24	GND
25	GND	26	GND
27	GND	28	GND
29	GND	30	GND
31	GND	32	GND
33	GND	34	GND
35	GND	36	GND
37	ERIDB(-) Read bus cmd	38	GND
39	REQ1(-) service Request 1	40	GND
41	REQ2(-) service Request 2	42	GND
43	SFLG(-) set complete flag	44	GND
45		46	GND
47		48	GND
49	WIDB Write strobe	50	GND

TABLE 2-3

Test CRT Connection Pinout

PIN	SIGNAL	DESCRIPTION
1	Chassis	Chassis ground
2	TD	Transmitter data ouput to CRT
3	RD	Receiver data input from CRT
7	GND	Signal ground

CHAPTER 3

OPERATION

3.1 GENERAL

Operation of the PDI is divided into three distinct areas. The first involves the power turn on and off procedures. The second concerns operation of the unit in a standalone mode using DIP switches and LED indicators located inside the unit to run and evaluate the diagnostic tests. The third area involves operation of the PDI as controlled by the computer for the establishment of a link and the transmission/reception of packet data.

3.2 POWER TURN ON/OFF

Under normal operations, power is applied to the unit using the front panel power switch. However, the operating mode of the unit is established by the 68000 processor during its powerup routines as either Diagnostic or Operational depending on the position of DIP switch number 4. This switch must be in the CLOSED (ON) position to powerup in the Operational mode. To change the DIP switch, the unit must be withdrawn from the rack and the top cover removed.

There are no special turn off procedures.

3.3 DIAGNOSTIC TESTS

In the event of problems, diagnostic routines are included in the on board firmware. These diagnostics are activated only after power up when the internal DIP switch located on the board has switch #4 in the OPEN/OFF position. Opening SW-4 while power is on will not stop the normal operation and begin diagnostic testing. Some tests may be run with no external devices connected. The tests requiring external equipment describe the functions and procedures required.

3.3.1 Operation

Test operation is controlled as follows;

1. Close SW-1 thru SW-8
2. Open SW-4
3. Apply power
4. Select test using SW 6,7, and 8
5. Close SW-4 to begin test
6. Open SW-4 to stop test

Additional tests may then be run by repeating steps 4 through 6.

Test selections;

0 -- Test LEDs (these report subsequent test results)

1 -- RAM test

2 -- ROM test

3 -- Asynchronous interface test

4 -- Host CPU port test

5 -- Miscellaneous register test & dialup test

SW-1 ON/Closed rotate a bit in host control register

SW-1 OFF/Open 1) try to dial # 657-5273

2) wait for operator to open SW-2

notes:

a. if SW-2 is off & SW-4 remains on, the test will retry to dial continuously

b. for single pass, close SW-2 before the test. Then at test end, open SW-4 to return for new test select & toggle SW-2 on-off-on.

6 -- 1 millisecond timer test

7 -- WD2511 loop back test

SW-1 & 2 off = WD2511 internal loopback

SW-1 only on = PDI loopback

SW-1 & 0 on = Modem loopback (no dialing)

The status of the test being run is presented by an 8 position LED assembly mounted next to the DIP switch. The LEDs display the following information;

DS1 - Power up diags passed

DS2 - Selected Diagnostic test passed last attempt

DS3 - Selected Diagnostic test failed last attempt

DS4 - Diagnostic test select enabled

DS5 - Failure Description code :

Test Number	Display	Meaning
1	1 (DS5)	Phase 1 read error
1	2 (DS6)	Phase 3 read error
2	1 (DS5)	ROM 1 checksum error
2	2 (DS6)	ROM 2 checksum error
3	1	Tx1 buffer never ready
3	2	Rx1 buffer never ready
3	3	Rx1 data incorrect
3	4	Rx1 error (framing or overrun)
3	9	(Tx # 2 errors)
3	A	-- same order as # 1 above --
3	B	
3	C	(these are encoded in DS5-8)
6	1	no 1 msec. flag received
6	2	1 msec. expired too quickly
7	1 (DS5-8)	can't establish linkup
	2	no PKR intrpt---timeout or XBA
	3	no XBA intrpt---timeout
	4	no flags detected
	5	regs. bad after data xfer
	6	TLOOK bad "
	7	RLOOK bak "
	8	(unused)
	9	RCNT not equal TCNT
	A	Recvr residue not 0
	B	Data transferred with errors
	C	Error interrupt rec'd from 2511
	D	No PKR intrpt -- after XBA

3.3.2 Test Descriptions

0) Test LEDS : Rotate a bit through each of the LEDs in sequence.

1) Test RAM : 3-phase Read/Modify/Verify test, using 32 walking data patterns;

```
0000 0000 0000 0001
0000 0000 0000 0011
0111 1111 1111 1111
1111 1111 1111 1111
1111 1111 1111 1110
1000 0000 0000 0000
0000 0000 0000 0000 (end)
```

For each pattern, test every RAM location as;

Phase 1: verify previous pattern

Phase 2: write new pattern

Phase 3: verify new pattern

2) ROM test: Checksum test

3) Asynchronous Port test: Loop channel 1 output to channel 1 input to channel 2 output to channel 2 input. Transmit all 256 8 bit patterns and verify at each point.

4) Host Port test: Output a walking data pattern to the host port (unconnected) and read back at input and verify.

5) Miscellaneous Register test: Output a rotating bit on the host control register for verification by operator using oscilloscope. For Dialup test, attempt to dial a number via the modem (657-5274). The operator must verify the number was called.

6) 1 Millisec. timer test: Using a software timing loop, enable and verify the 1 millisec. countdown timer circuit operation.

7) WD2511 Loopback test: Loop the transmitter section of the packet controller chip to the receiver (internal, external on the PDI board, or external through the modem in loopback mode) and verify the ability to establish a link and transfer data packets.

3.4 OPERATION WITH HOST CPU

All operation of the PDI is under control of the host CPU. This is accomplished using a parallel data path between the host (HP computer) and the 68000 control processor within the PDI. A device driver is supplied for the PDI which supports this path. The following section details the operations available to the host software to enable it to implement higher levels of the OSI model.

3.4.1 Communications

All communications with the PDI utilizes byte block transfers. The blocks are read or written using standard system EXEC routine calls. The driver installed in the system is concerned only with the block transfers. It will return error status when a transmission error, such as parity or checksum error, occurs.

Block identifiers are included within the blocks. All blocks conform to the following format:

byte 0 : Operation Code
bytes 1-n : Data

Parity and checksum generation and checking are performed by the driver software and is transparent to the user.

Three EXEC calls are supported,

EXEC call #1 : Read from PDI

EXEC call #2 : Write to PDI

EXEC call #3 : Clear interface with PDI

3.4.2 Formats

The format for read and write EXEC calls is;

```
CALL EXEC(ICODE, ICNWD, IBUFF, ILEN)
```

The format for the I/O control EXEC call is;

```
CALL EXEC(ICODE,ICNWD)
```

where ICODE is the request code

1 for Read, 2 for Write, 3 for I/O

ICNWD is the control word

IBUFF is the data buffer

ILEN is the length of the data buffer

(in words or -bytes)

The control word format (ICONWD) is;

```
 15 14 13 12 11 10  9  8  7  6  5  4  3  2  1  0
-------------------------------------------------
|0  0  0  0  0 |  fcn code    |    LU #          |
-------------------------------------------------
```

Read function code : 00 - Read requested data block from PDI

Write function code: 00 - Write data block to PDI

I/O control code: 00 - Clear host/PDI interface

3.4.3 Operation Codes in Write EXEC Call Blocks

00 - initialize PDI to power up state

block contents : 2 bytes of zeroes

01 - write data block to transmitter block queue and output

block contents : 2 - 1024 bytes of packet data

Note: each block transfer represents a complete packet

02 - Write data block to configuration buffer and reconfigure accordingly (includes dialing if DTE = 0)

```
block contents :  bit  7   6   5   4   3   2   1   0
                      -------------------------
                      |    D-A     |    D-B     |
                      -------------------------
```

byte #	D-A Contents	D-B Contents
1	long distance 1/0	DTE/Loop/Call/Hangup
2	area code, MSD	area code, NMSD
3	area code, LSB	Exchange, MSD
4	Exchange, NMSD	Exchange, LSD
5	Number, MSD	Number, NMSD
6	Number, NLSD	Number, LSD

Notes: 1. byte 1, D-B contents;
bit 0 : DTE select bit
1 : Local loopback select
2 : Call(1) / Hangup(0) select
2. to suppress a long distance 1, enter zero
3. to suppress area code, enter zero in all 3 bytes
4. DTE=1 indicated the PDI should act in passive mode, may not initiate linkup or dial remote site

03 - Read Data request

block content : 2 zeroes

04 - Read Status request

block content : 2 zeroes

3.4.4 Status Block

Byte 0: Bit 0 - Illegal command received

1 - TX buffer not empty

2 - No configuration data received

3 - Hardware failure detected

4 - Link is up

5 - Timeout on outstanding blocks (no ACK from remote)

6 - Read buffer empty

7 - Link was downed with outstanding TX blocks

3.4.5 Link Operations / Procedure

The following summarizes operation across the HOST/PDI interface for an initiating station:

a. Execute an initialize operation to ensure the PDI & link are in a known state.

b. Execute a configure operation to establish a physical circuit and virtual data path.

c. Execute a series of write data blocks to transfer data packets to the remote site.

d. To receive data packets, first read the status block. When the receiver buffer becomes not empty, execute a read packet request. Reading the status block after each packet is read will indicate whether more data has been received.

e. Periodically, before 'c' & 'd' especially, examine the status block.

3.5 PDI OPERATIONS

This section describes the operations performed by the PDI.

3.5.1 Initialize

- ensures modem is on-hook
- initializes the TX and RX buffers to empty
- clears previous configuration data
- sets the hardware & software to the power-on state

  (This means the status word will NOT show the link downed with outstanding blocks)

3.5.2 Configuration

The configuration block transfers provide dialing, linkup, and linkdown operations.

3.5.3 Dialup/Linkup

When the linkup bit is set in the first byte of the block, the following sequence of events occur:

1. The level 2 controller addresses are set as the DTE or DCE
2. If the modem is already online, the command is ignored and the Illegal command status is posted.
3. An attempt to dial the number supplied is made.
4. All TX and RX data buffer pointers are reset to the first block.
5. If the PDI was configured as the DTE, an attempt to linkup (SABM)

is made. If it fails after N2 attempts, all is reset to the power up state.

6. The LINKUP status is posted.

7. END

3.5.4 Disconnect / Linkdown

When the linkup bit is reset in the first byte of the block, a mandatory disconnect is issued by the level 2 controller chip. The controller will issue a DISC and down the link. The TX and RX data buffers are initialized (data is lost). After either an acknowledge from remote or timeout error, the controller will be reset to its power up state and the modem placed on-hook.

The status block will accurately report status as affected by the link down operation. Due to possible delay, however, the status will reflect the down conditions only when the linkup bit is reset.

3.5.5 Sending Data

The send data (TX) block transfer adds data to the send queue (up to 7 outstanding, unacknowledged packets supported) and instructs the controller to attempt to send it.

1. TX buffer not empty status is posted.
2. The controller is commanded to send the block.

Startup errors will be posted in the status word. Fatal transmission and link errors will be posted and additionally the controller is initialized and the modem placed on hook. Therefore, if multiple blocks are outstanding at the time of error, the host has no way of knowing which blocks were acknowledged and which were not.

3.5.6 Receiving Data

Once the link is up, data received will be placed into a circular buffer, (7 packets of up to 1024 bytes long). The controller will detect data overwrite if the host does not empty the data and will refuse further input.

To receive data packets, read the status block. When the RX buffer becomes Not Empty, execute a read RX data request. Reading the status block after each packet is input to the host will indicate whether more data has been received.

If an RX data block request is made and no data exists in the buffers, the status is updated to reflect this condition (Illegal command, Read Buffer Empty) and no data or hardware reset or initialize occurs.

If during the read transfers, the HOST-PDI interface experiences a transmission fault, the read request may be restarted and the data will be read from the beginning of the block.

Link errors that cause the controller and modem to be initialized will also cause the receive buffers to be lost with no indication of the amount of data lost. This includes errors which are detected during transfer of a buffer to the host.

CHAPTER 4

CIRCUIT DESCRIPTION

4.1 GENERAL

This section describes the PDI in three levels. First, the overall unit block diagram is reviewed to identify the major components and their function. Next the circuit board's detailed block diagram is reviewed to illustrate its major functions. Third, a page by page logic diagram description identifies where block diagram elements are located and where required, a detailed design description if provided.

Where a number appears within a block diagram, it refers to the schematic page number where the function is found.

4.2 BLOCK DIAGRAM DESCRIPTIONS

4.2.1 Overall Unit Block Diagram

Figure 4-1 presents the PDI block diagram. There are two major assemblies, the circuit board and the power supply. Other than the power switch, there are no front panel controls or indicators. The circuit board provides 4 interfaces. All unit operation is controlled from the Host CPU using a 16 bit parallel interface. All serial data communications to the Group 4 terminal under test is made though an RS-232 serial interface to a modem.

For test and maintenance purposes, two additional RS-232 ports are provided as well as on board diagnostic test control and indicators.

4.2.2 Circuit Board Block Diagram

Figure 4-2 presents the block diagram of the circuit board. It shows the MC68000 bus as the only data path within the unit. All data and control information is passed from the host to the RAM buffer areas under direct control of the MC68000. The 68000 also controls the operation of the packet data controller (WD2511 I.C.). This chip passes packet data to and from the RAM and serial link using Direct Memory Access techniques. The 68000 monitors the WS2511 status, setups the DMA buffers, and provides the overall link control while the 2511 provides all the bit-oriented control and processing to implement LAPB.

The memories consist of 16K x 16 bits of UV-Erasable PROMs, and 64K x 16 bits of Dynamic RAM. The large RAM supports the multiple transmit and receive packet data buffers as well as program variables and the 68000 stack.

The two serial ports are provide by a pair of type 6850 Asynchronous Communications Interface Adapter I.C.s.

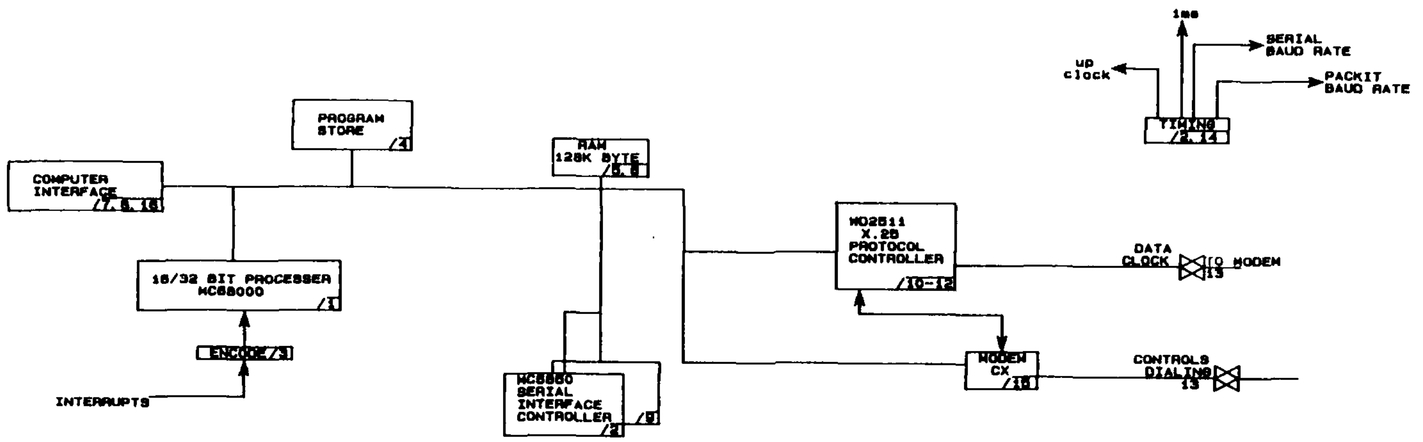


FIGURE 4-2 CIRCUIT BOARD BLOCK DIAGRAM

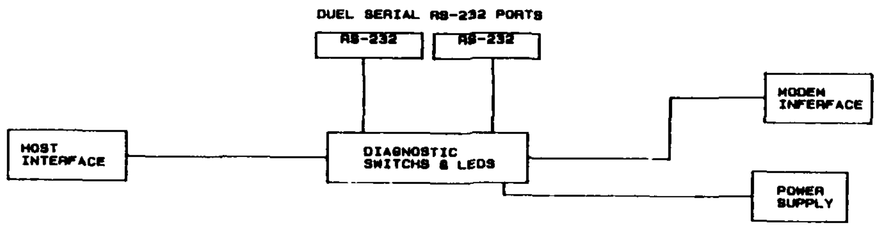


FIGURE 4-1 PDI UNIT BLOCK DIAGRAM

Size	Document Number	REV
B		
Date: April 21, 1987	Sheet of	

Modem control consists of the dialing control and flow control (Clear to Send, Request to Send, etc.) These signals are intercepted by registers for 68000 processor control. This is required to support half duplex operation. The processor can therefore negate the CTS to the 2511 packet controller to throttle data output.

5.3 LOGIC DESCRIPTION

Sheet 1, Processor and Bus Buffers

The processor is run at 8 Mhz. The address and data bus is buffered due to the large fannout required.

Sheet 2, Address Decode & DTACK Generate

Decoders U21 and U22 decode the 68000 memory space into the following segments;

00 0000 - 00 7FFFh	EPROM, program store
01 0000 - 01 001Fh	Packet Controller Registers
01 0020h	Modem Control Register
01 0030h	Host Interface Data Port
01 0040h	Host Interface Control Regs.
01 0050 & 01 0052h	ACIA #1 Control & Data Regs. 02
01 0060 & 01 0062h	ACIA #2 "

01 0070h	Diagnostic Switches & LEDs
0000 - 02 FFFFh	RAM

U15 generates a fixed delay DTACK (Data transfer Acknowledge) signal for the memory or registers which do not provide a busy indication.

Sheet 3, Timing Generation and Interrupt Logic

The logic on the left hand of this sheet provides the basic timing signals for the processors (68000 & WD2511) and the serial data transfer baud rates. Encoder U14 converts the interrupt requests into prioritized levels for input to the 68000. Autovectoring mode is used in this design, hence gate U18-10 generates VPA in response to the Interrupt Acknowledge bus operation.

Sheet 4, PROM

Two UV-Erasable PROMs are arranged in a word wide configuration, providing 16K by 16 bits of program storage.

Sheet 5, Dynamic RAM Controller

The type TMS4500A controller chip converts the 16 address bits of the bus into multiplexed row and column data as required by the

DRAMS (sheet 6.) This chip also determines the memory access timing and therefore gates U19 and U50-3 use the CAS signal (Column Address Strobe) to generate the DTACK. DRAM accesses may come from either the MC68000 or the WD2511.

Sheet 6, Dynamic RAMs

Sixteen 64k x 1 RAM chips are arranged in a parallel fashion to provide the 64K x 16 bit RAM memory.

Sheet 7, Test Loop Mux & Host Interface Control

Multiplexer chip U70 is used to control the source of three signals. The signals are the Host Interface driver enable (ERIDB) and the two ACIA receiver inputs. The sources are changed depending on the state of the LOOPA signal which is asserted in the test modes to permit the ACIA transmitter output to be routed into the ACIA receiver inputs. The host interface drivers are constantly enabled in the test mode to allow the readback of the host output data.

The host interface logic converts the 68000 bus signals into the proper polarity and pulse width that is required by the host interface card.

Sheet 8, Host Interface & Miscellaneous Control Register

This register controls the data transfers over the PDI/HOST parallel interface by resetting the interrupts received (RI[n]) and signalling the host data is available (SFLAG). The PAK and LOOPA signals control the routing of packet and ACIA data outputs for loopback testing operations.

Sheet 9, ACIA Serial Interface

Two type MC6850 asynchronous controllers provide serial ports for test purposes.

Sheet 10, WD2511 Read/Write Control

This sheet contains the timing logic to permit the 68000 to access the WD2511 control registers. The read and write signals for the WD2511 must be asserted at much longer times after the address and data appear than the 68000 bus normally operates. Therefore U65 generates signals at 1.4 microsecond intervals which are used to derive the WD2511 read and write signals, PACRE & PACWE, as well as the delayed DTACK. The WD2511 is an eight bit device, so all transfers are made in byte mode. This requires the upper and lower data bytes of the 68000 bus to be alternately enabled to the 8 data bits of the WD2511 (U52-8,11 and U18-11).

Also on this page, U49 provides the upper address bits during the WD2511's DMA transfers with RAM.

Sheet 11, WD2511 DMA Control

This logic provides all the bus mastership timing for the WD2511. When the packet controller reads or writes memory, it first asserts the DMA Request In or Out signal (DRQI/O) which is passed to the processor as a Bus Request. When the Bus Grant is received and the previous bus transaction is complete (U58-8), the Packet Controller and this logic control the bus (PACBUS signal asserted). Shift register U66 provides the timing for the data & address to AS and UDS/LDS deskewing. Flip flop U67-9 is set after the Data Strobe is generated which suspends the timing shift register operation. Therefore, the control logic waits until the DTACK is received from the memory (gates U59-3 and U71-3).

Sheet 12, Packet Network Interface Controller

This sheet contains the W2511 controller I.C. Note the two bus buffers, U/2 and U73, connecting the eight bit chip to one of the two bytes of the 16 bit 68000 bus.

Sheet 13, Serial Data Drivers & Receivers

This sheet contains the TTL / RS-232 translators used for the packet interface and the two asynchronous ports.

Sheet 14, Flag Detector and 1 millisecond Flag

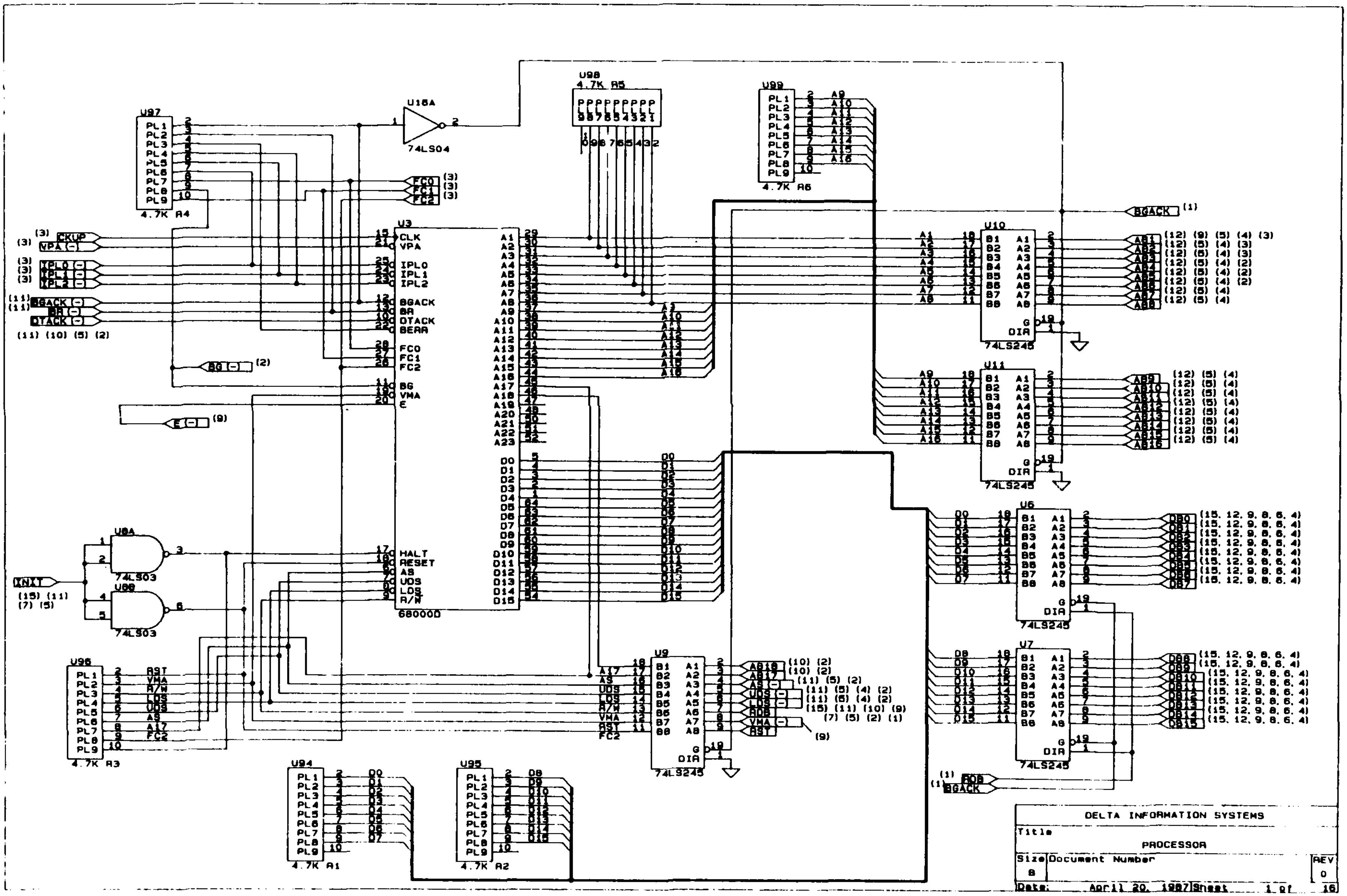
Shift register U80 and gates U17 and U18 detect the presence of a flag pattern at the receiver input. This supports the half duplex mode of operation when the PDI must communicate using a single 2-wire interconnect. The second logic section sets the 1 millisecond flag (MS1), also used to support half duplex operation. If enabled by MSEN, at the end of each 1 ms period, the counter (on sheet 3) is reloaded by LD1MS and the MS1 flag set. When read by the processor, it momentarily sets the RSTMS1 reset signal to reset the flag.

Sheet 15, Modem Command & Status Register and Diagnostic Test Control

U54 and U61 provide the 68000 the capability to intercept the flow control signals between the WD2511 and the modem in order to implement the half-duplex mode. U68 permits the 68000 to read DIP switches at U92 which control the diagnostic tests operations. U69 is the holding register for the diagnostic LED results displays.

Sheet 16, Host Interface Data Path

This is a 16 bit bi-directional data bus.



DELTA INFORMATION SYSTEMS

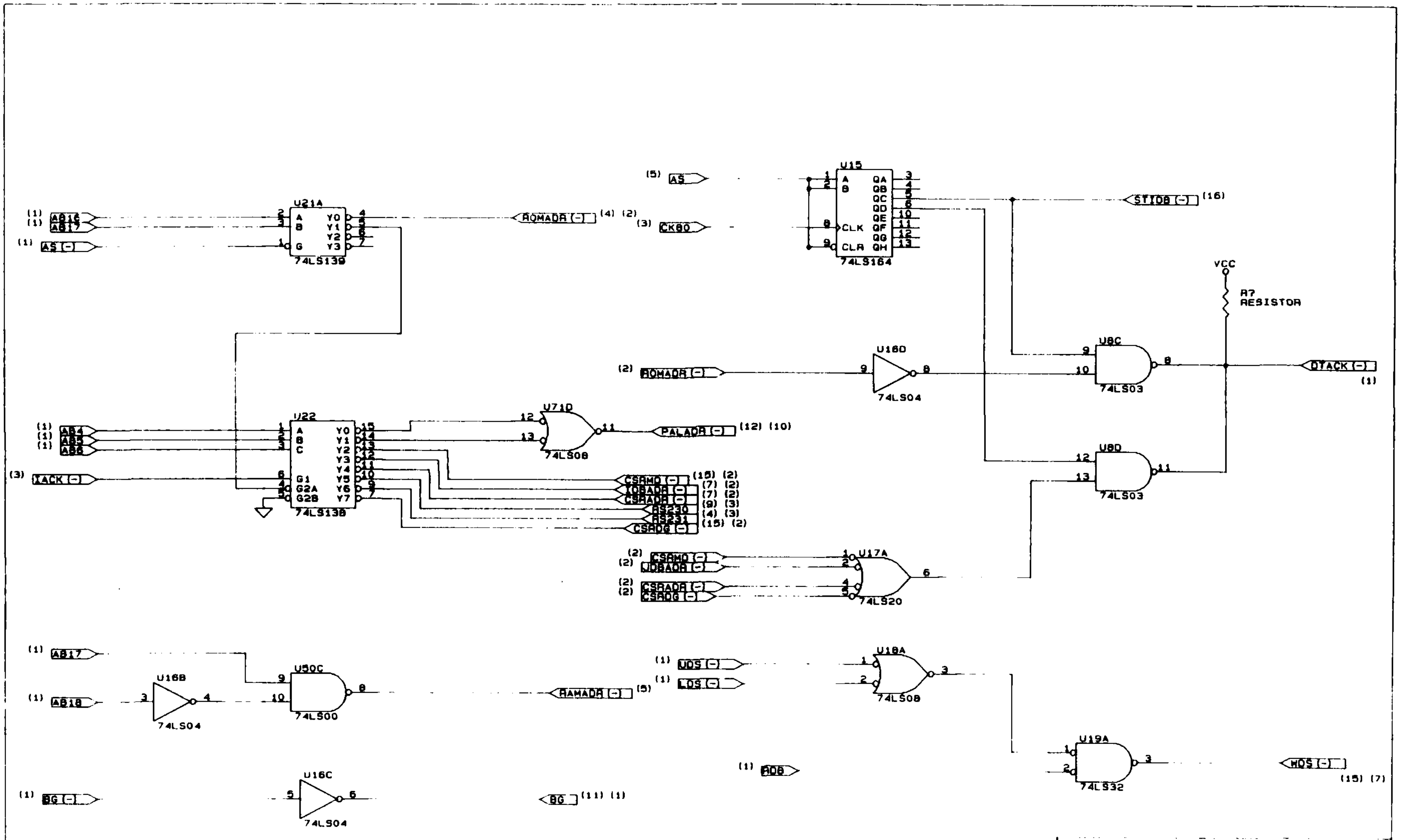
Title PROCESSOR

Size B

Document Number

REV 0

Date: April 20, 1987 Sheet 1 of 16



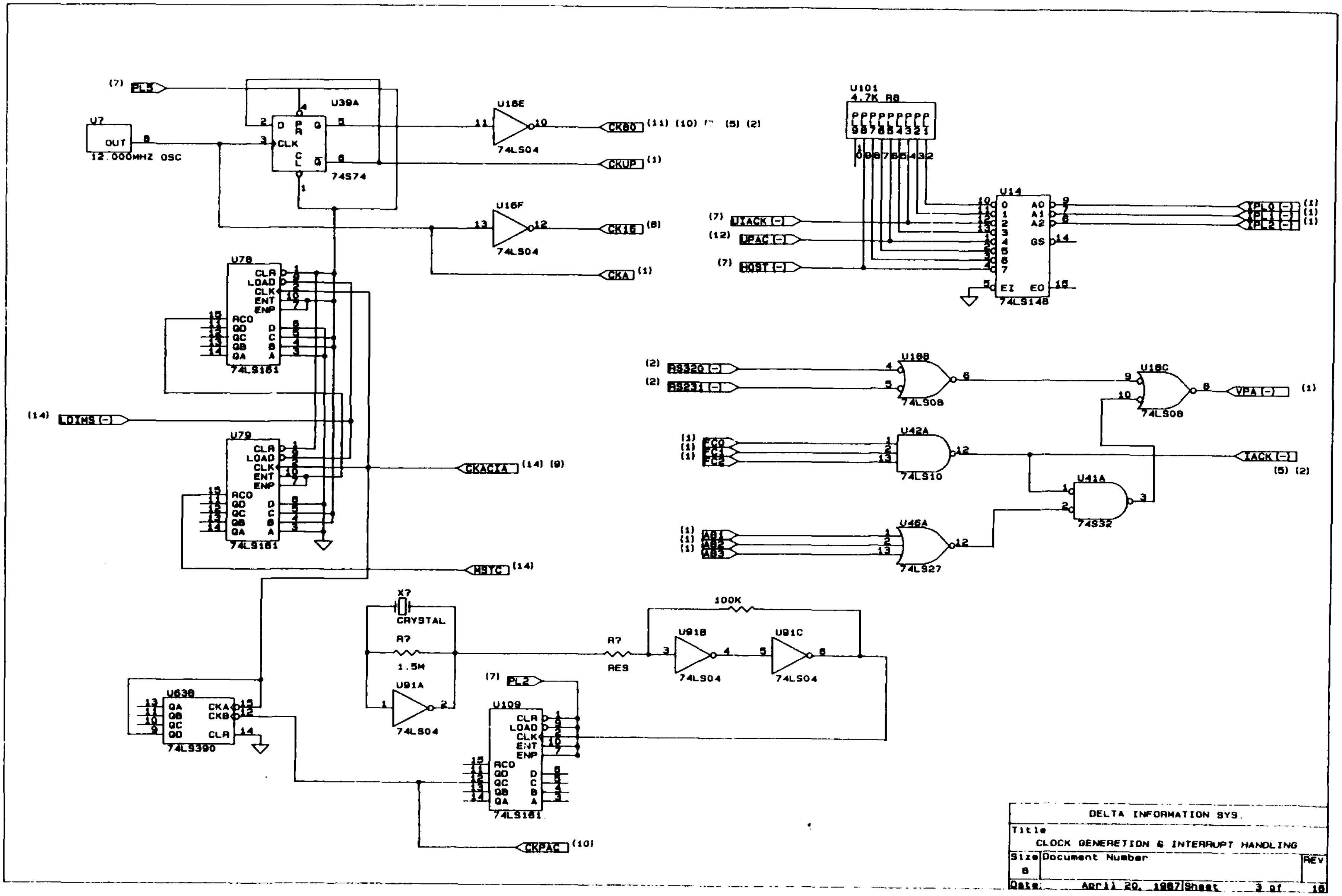
DELTA INFORMATION SYS.

Title

ADRS DECODE & DTACK GEN

Size B Document Number REV

Date: April 20, 1987 Sheet 2 of 18

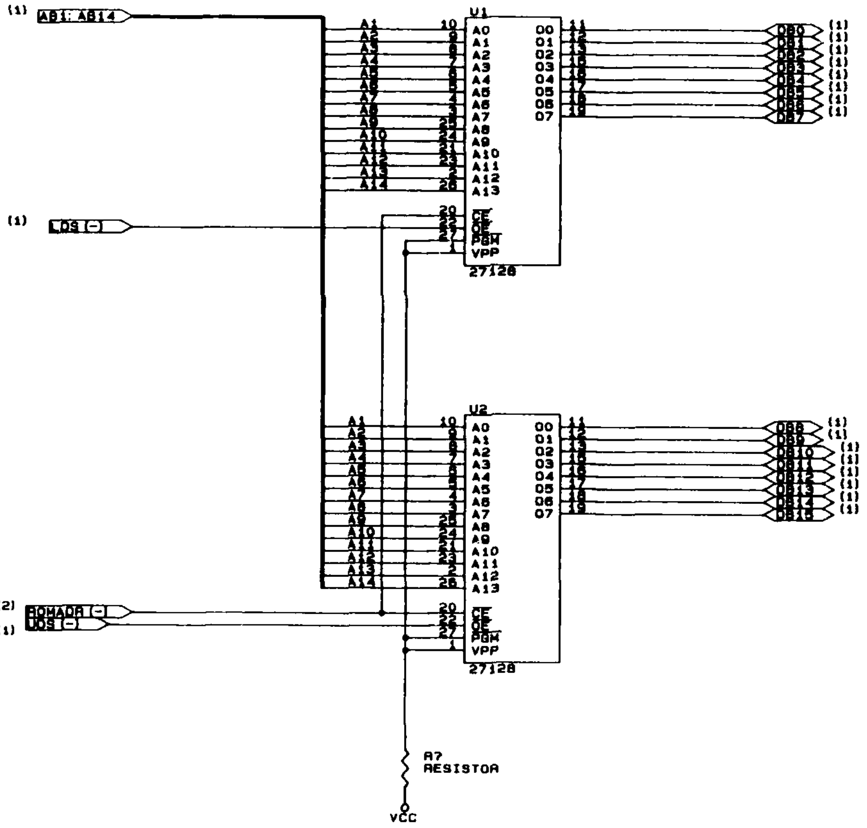


DELTA INFORMATION SYS.

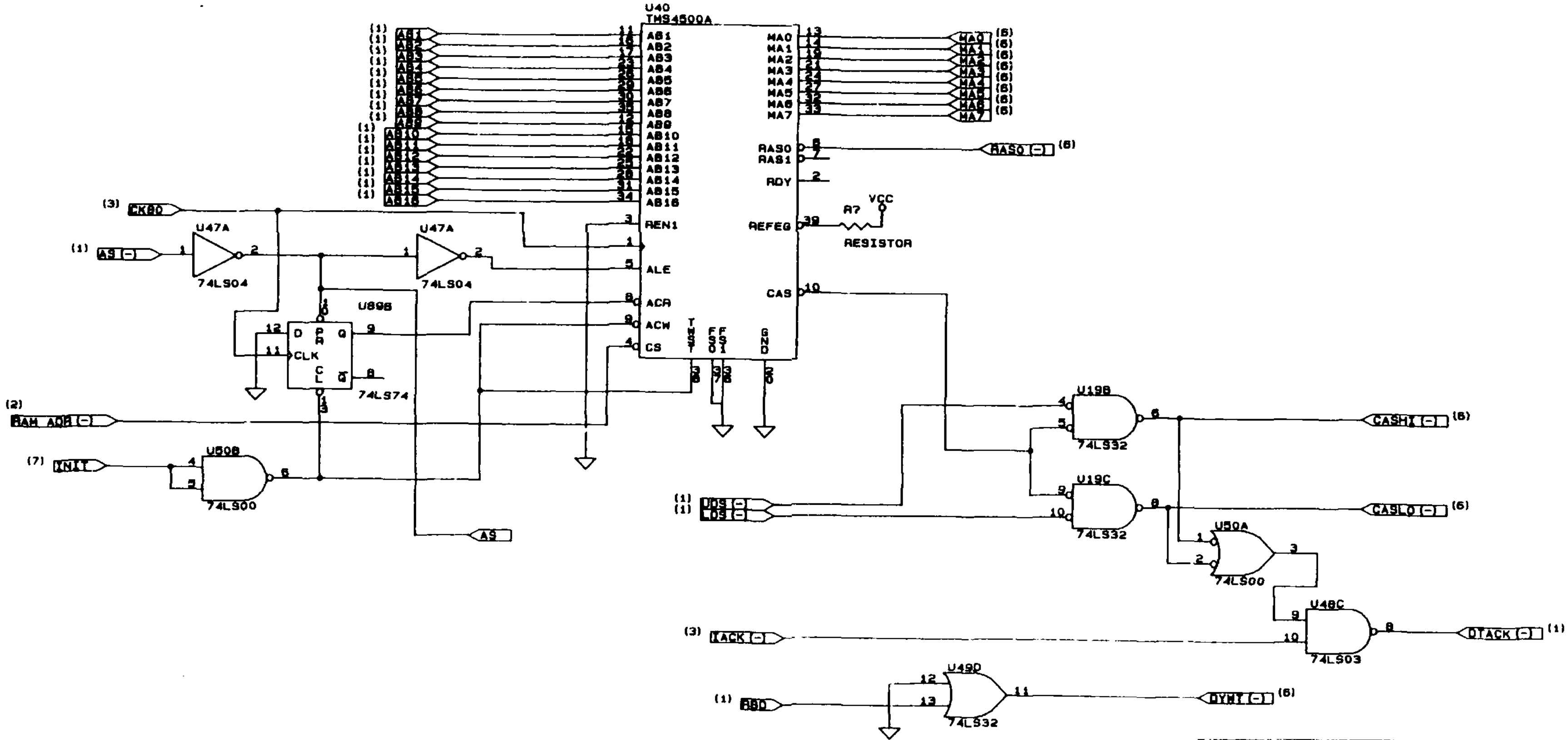
Title
CLOCK GENERETION & INTERRUPT HANDLING

Size	Document Number	REV
B		

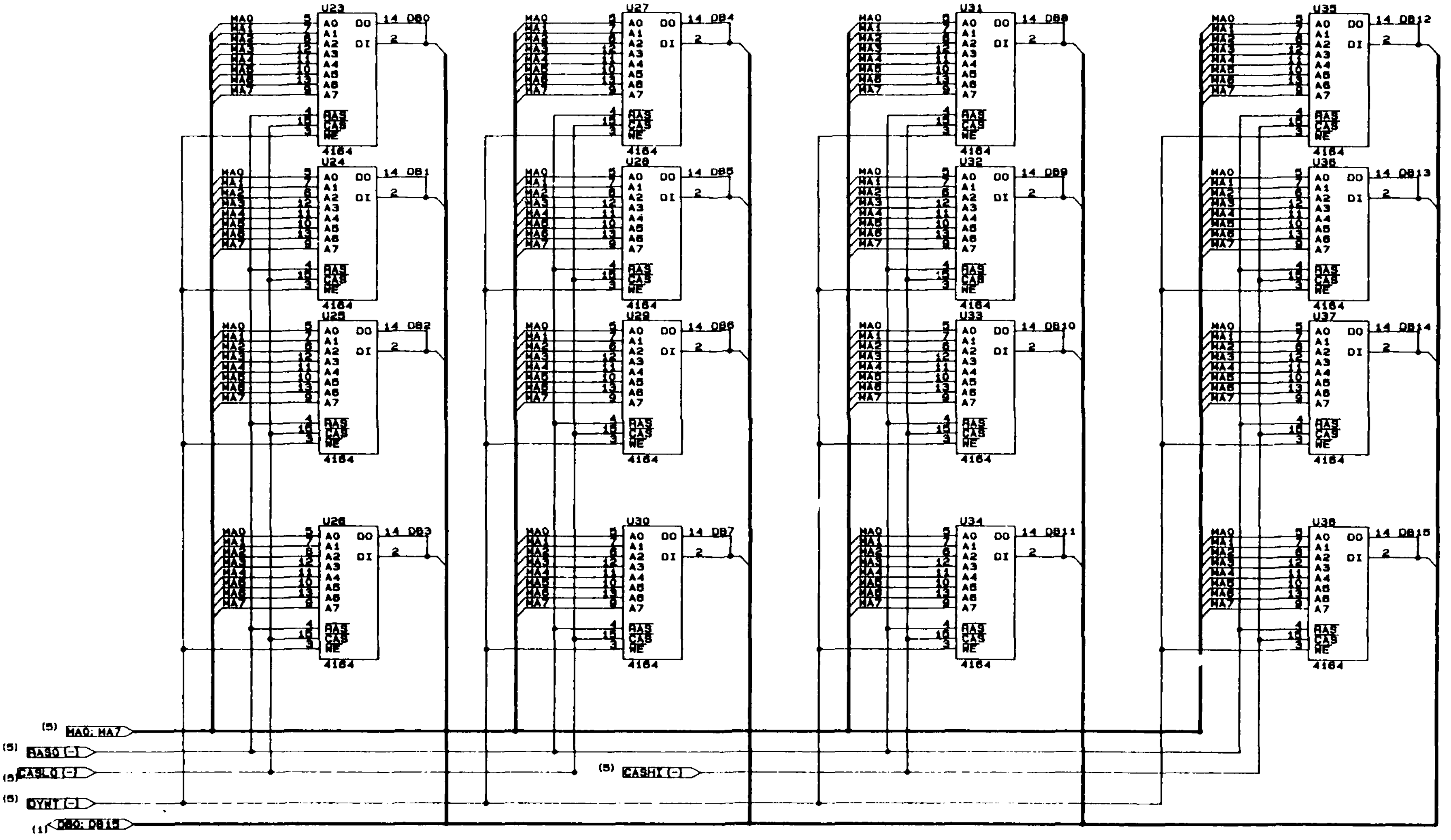
Date: April 20, 1987 Sheet 3 of 16



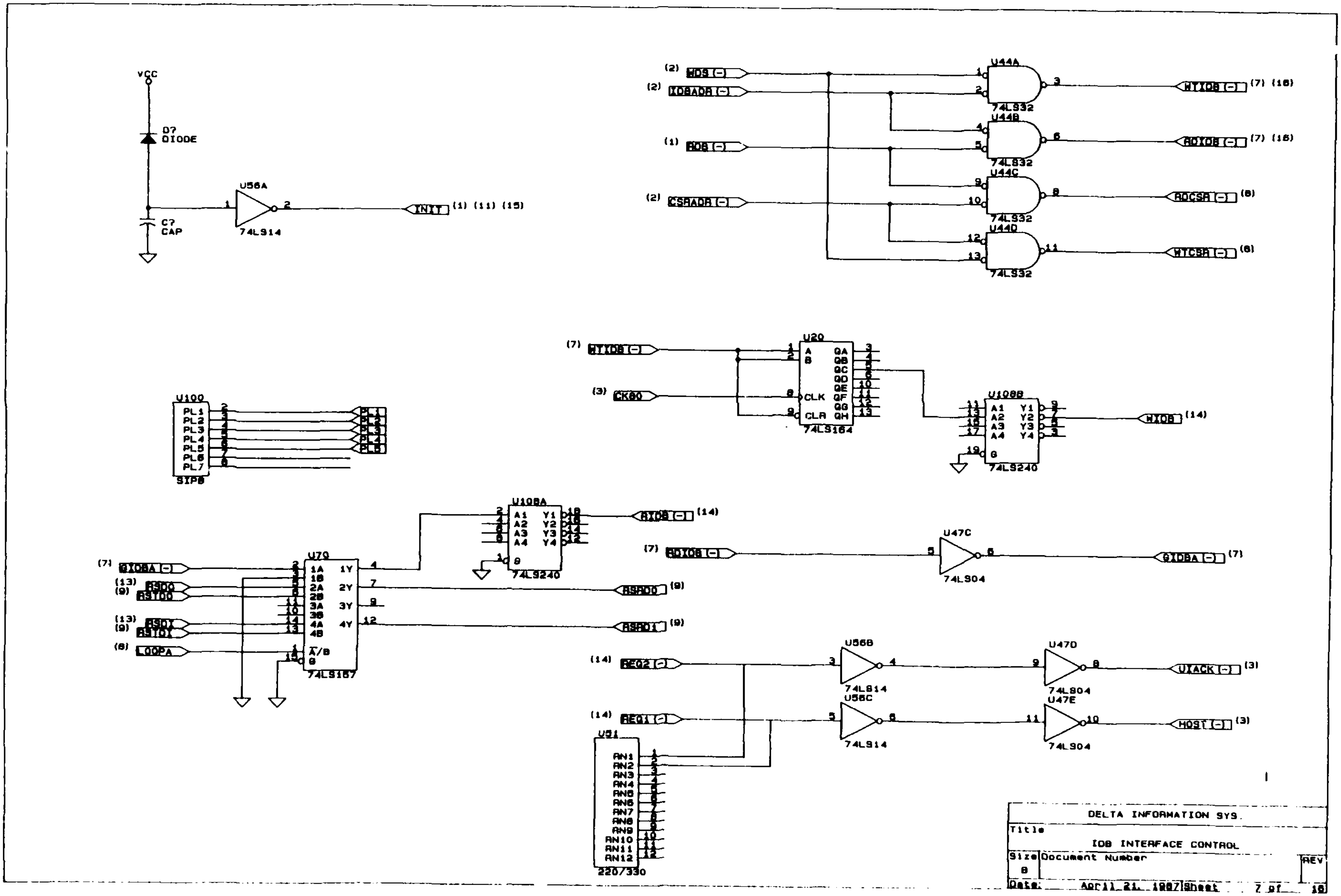
DELTA INFORMATION SYS		
Title PROM		
Size B	Document Number	REV
Date: April 20, 1987	Sheet 4 of 16	



DELTA INFORMATION SYS.		
Title DYNAMIC RAM CONTROL		
Size B	Document Number	REV
Date: April 20, 1987	Sheet 5 of 10	



DELTA INFORMATION SYS.			
Title	DYNAMIC RAMS (128K BYTES)		
Size	Document Number		REV
B			
Date:	April 27, 1987	Sheet 8 of	16



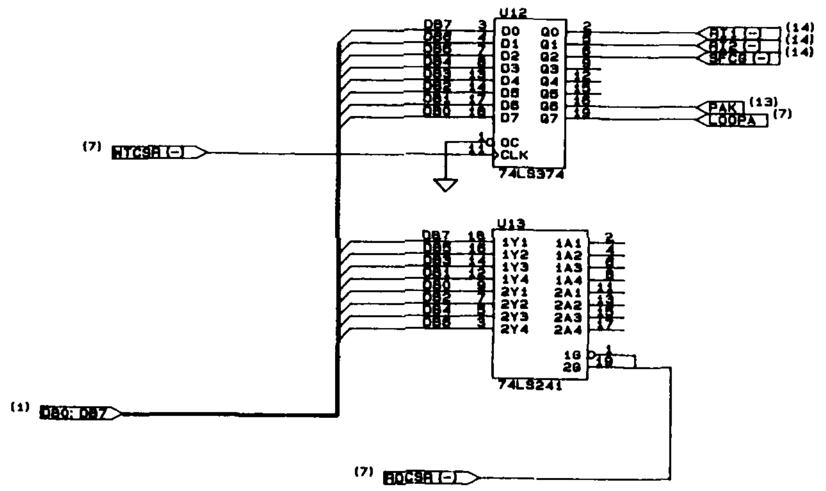
DELTA INFORMATION SYS.

Title

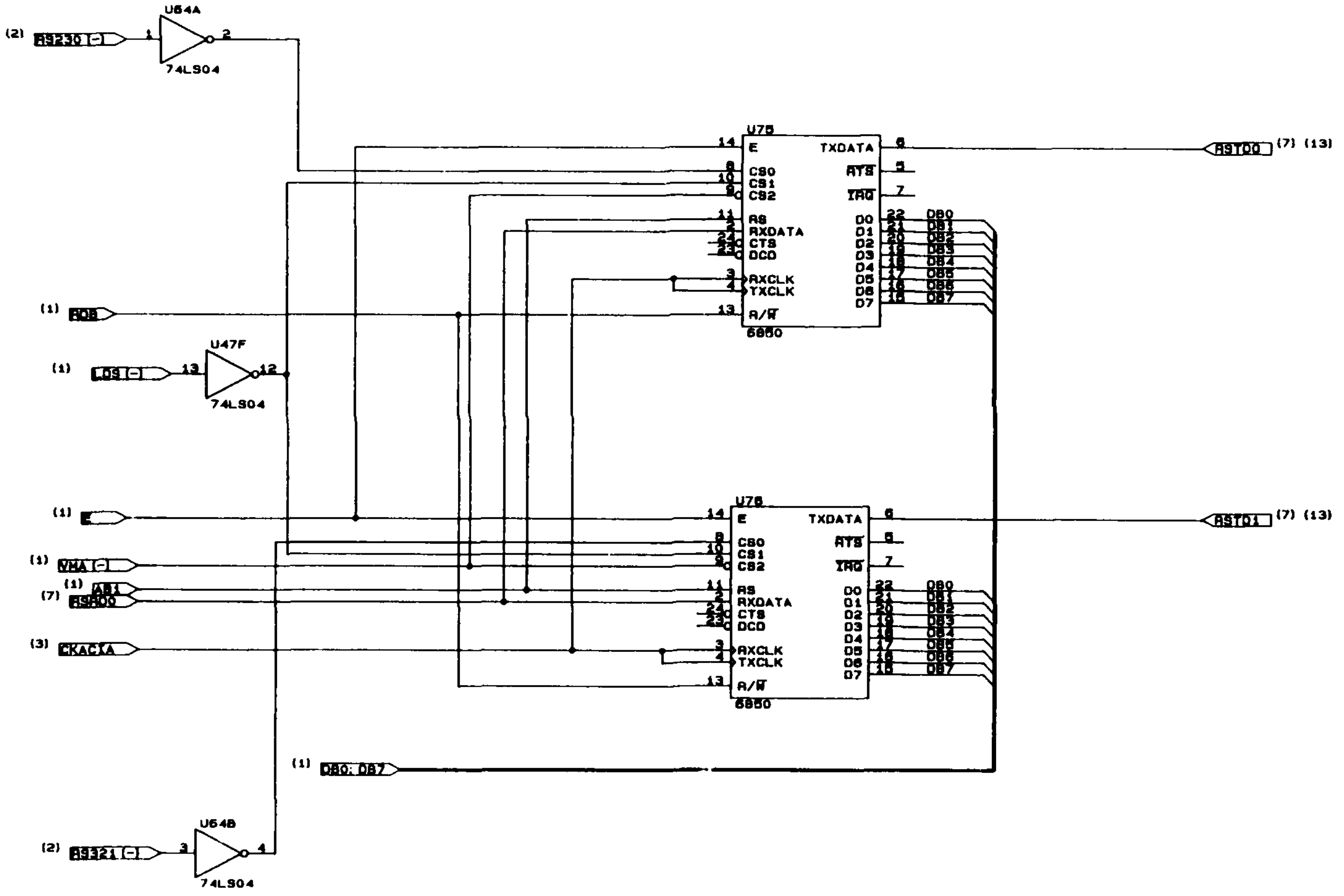
IOB INTERFACE CONTROL

Size B | Document Number | REV

Date: April 21, 1987 | Sheet 7 of 16

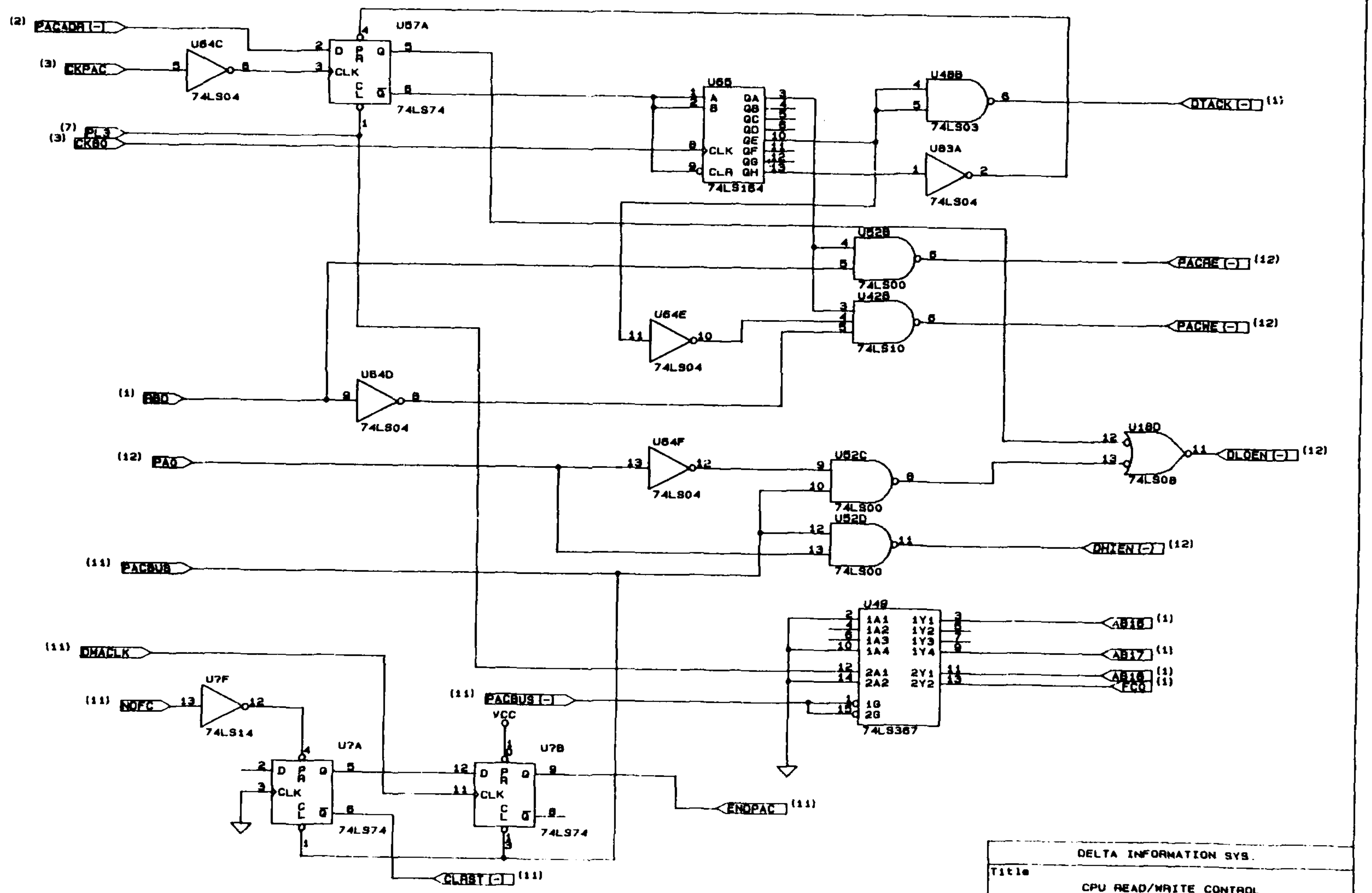


DELTA INFORMATION SYS.			
Title IDB PATH & CSR			
Size B	Document Number		REV
Date:	April 21, 1987	Sheet 8 of	16



DELTA INFORMATION SYS.

Title	ASYNC SERIAL INTERFACE	
Size	Document Number	REV
B		0
Date:	April 21, 1987 Sheet 9 of	18



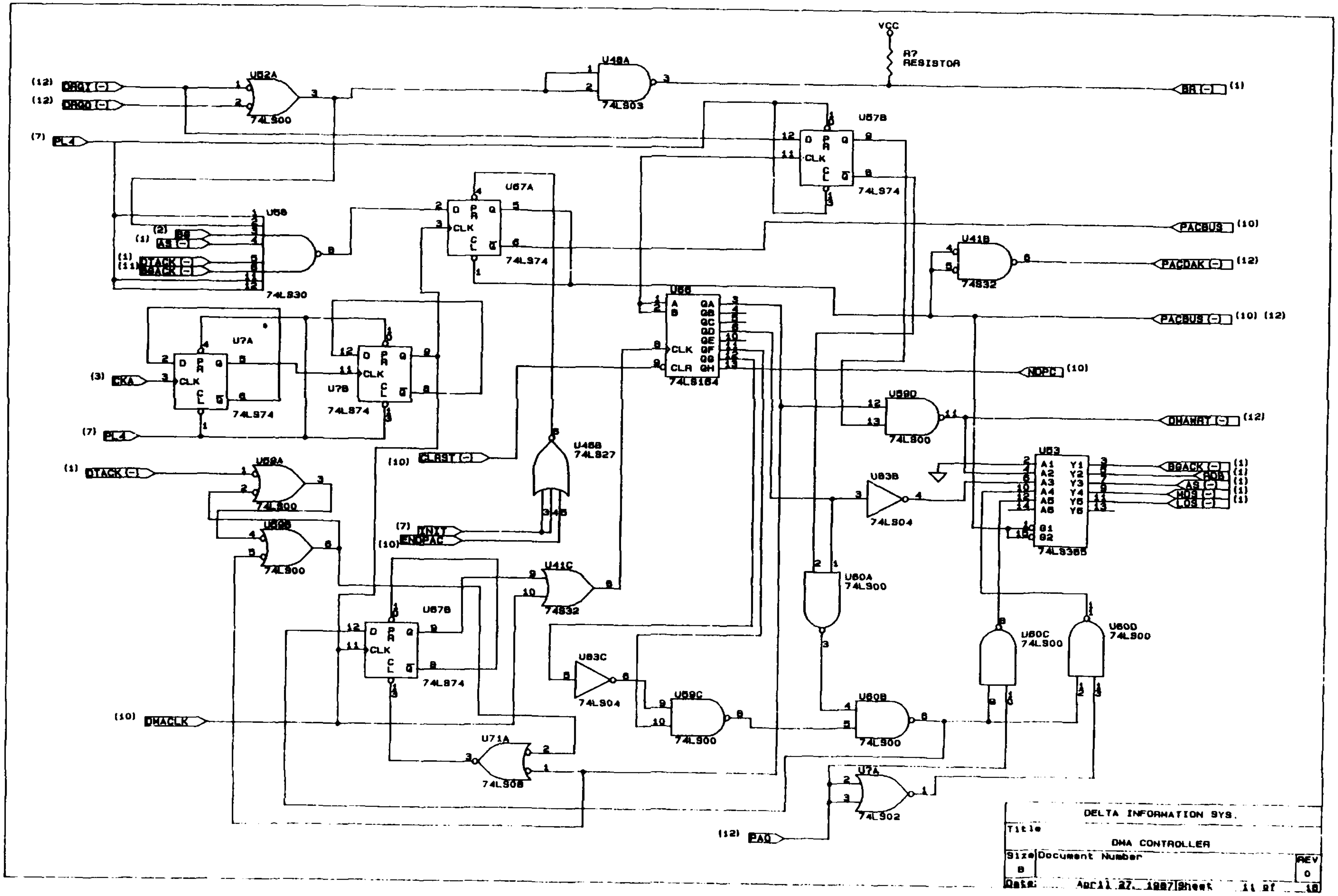
DELTA INFORMATION SYS.

Title

CPU READ/WRITE CONTROL

Size	Document Number	REV
B		0

Date: April 21, 1987 Sheet 10 of 16

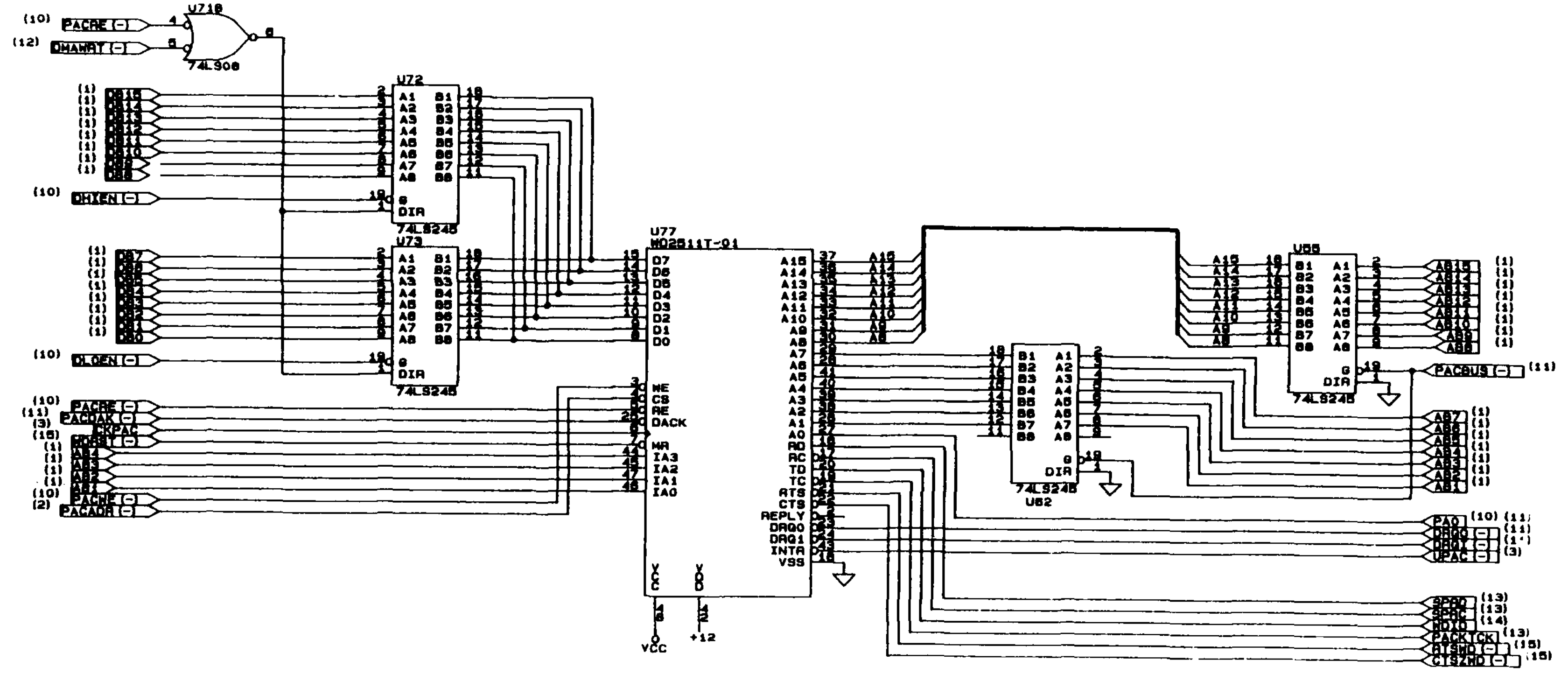


DELTA INFORMATION SYS.

Title DMA CONTROLLER

Size B Document Number REV 0

Date: April 27, 1987 Sheet 11 of 18



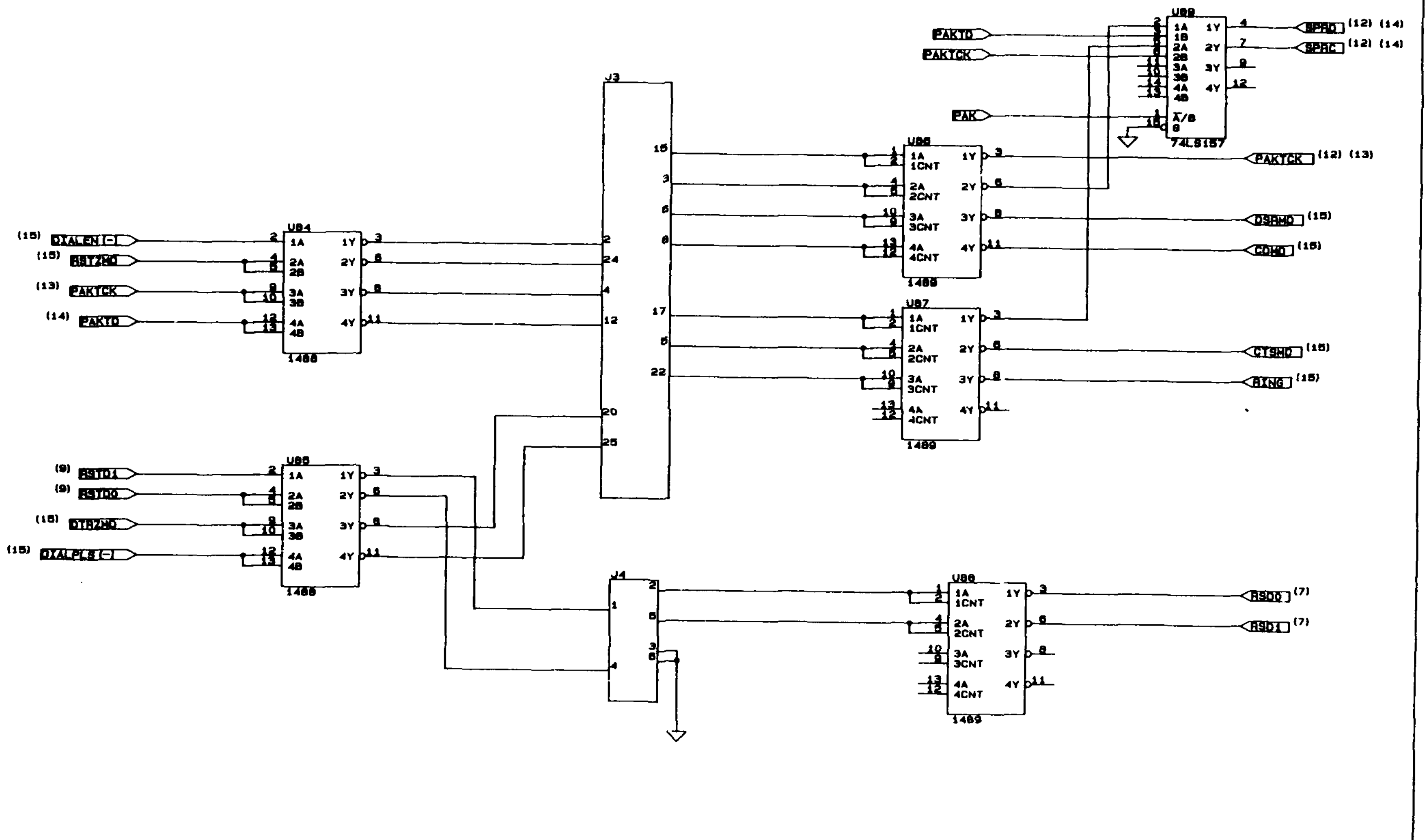
DELTA INFORMATION SYS.

Title

PACKET NETWORK INTERFACE

Size	Document Number	REV
B		

Date: April 21, 1987 Sheet 12 of 16



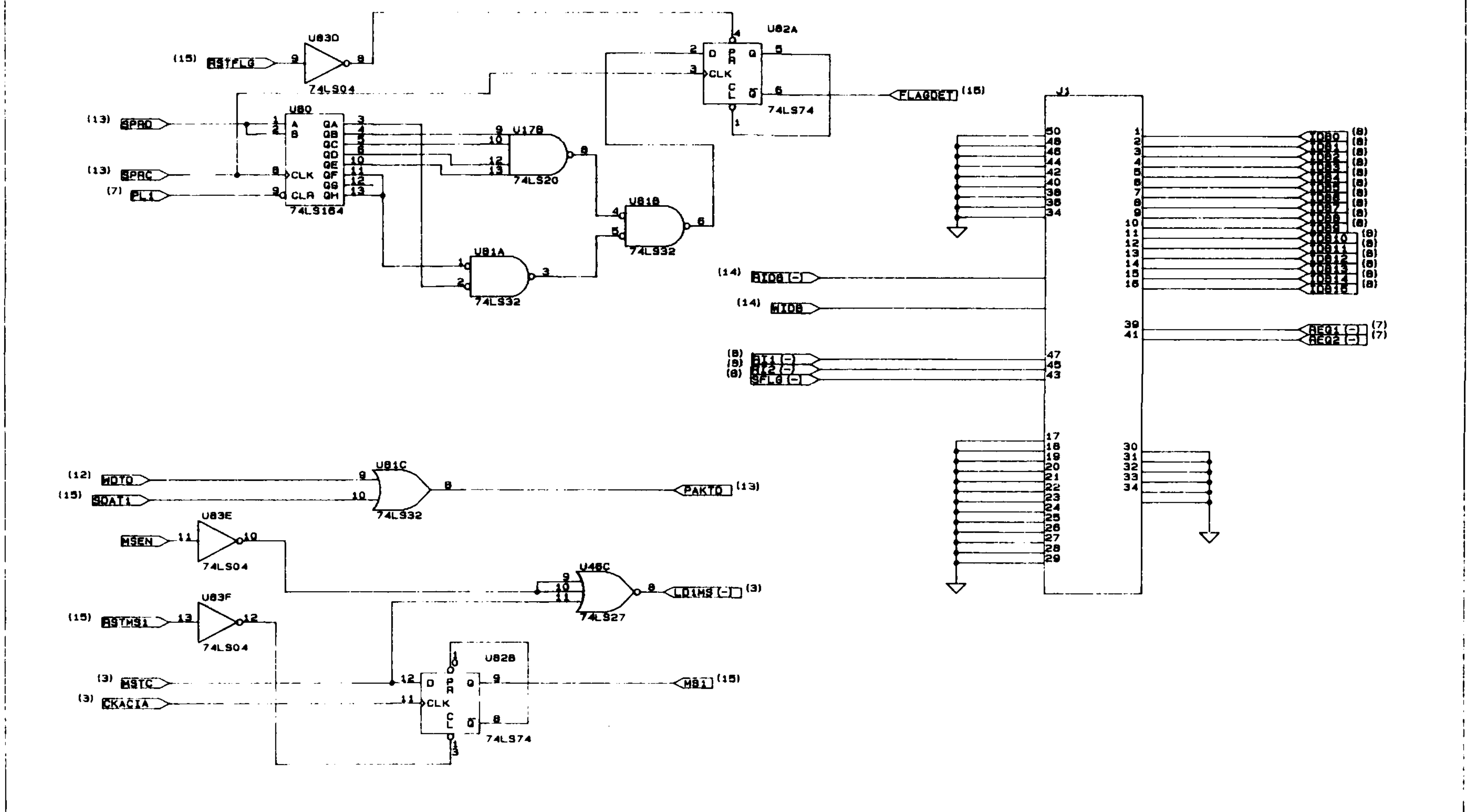
DELTA INFORMATION SYS.

Title

SERIAL DATA DRIVERS & RECEIVERS

Size B | Document Number | REV

Date: April 21, 1987 | Sheet 13 of 18



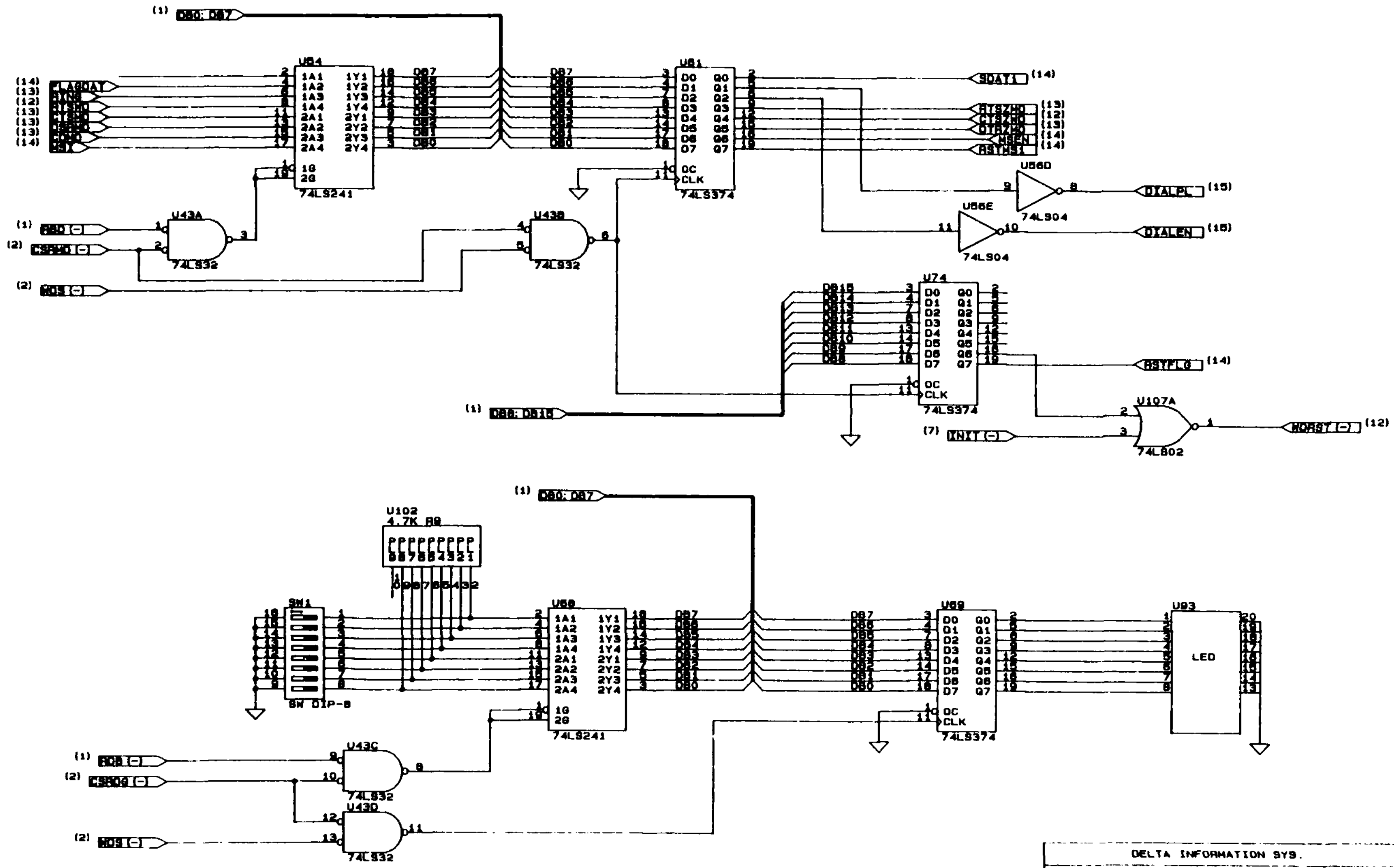
DELTA INFORMATION SYS.

Title

FLAG DETECTOR/1MS FLAG

Size	Document Number	REV
B		0

Date: June 9, 1987 Sheet 14 of 16



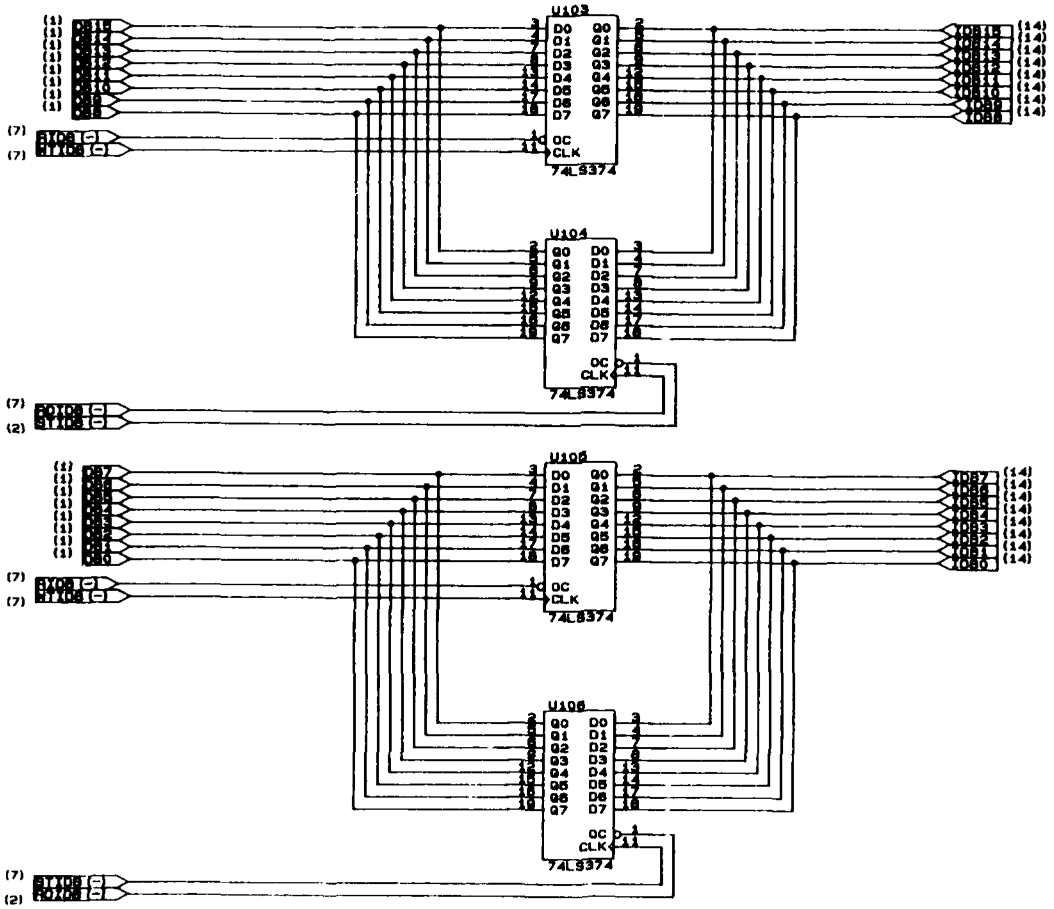
DELTA INFORMATION SYS.

Title

MODEM CSR & DIAG. TEST CONTROL

Size	Document Number	REV
B		

Date: April 21, 198/ Sheet 15 of 18



DELTA INFORMATION SYS.

Title

IDB DATA PATH

Size	Document Number	REV
B		0

Date: April 27, 1987 Sheet 10 of 10

APPENDIX B

VALIDATION SYSTEM SOFTWARE DESCRIPTION

Appendix B

B.0 Validation System Software Description

B.1 Software Design Approach

B.1.1 System Concept

Since the basis for Group 4 Facsimile equipment is the Teletex protocols as described in the CCITT recommendations listed in Table B.1, the validation system software's primary responsibility is the implementation and validation of these protocols. Along with this requirement, the validation system must also be capable of specifying and testing the different parameters/variables allowed within each of the protocol layers as defined in the recommendation for that layer.

Figure B.1 is a functional block diagram of the validation system software. From an overall point of view, the validation system drives two operations - the UUT and the Group 4 terminal emulator - and compares the results. The emulator acts as a "golden" model against which the performance of the UUT is compared, giving due allowance for permissible variations in operation. By substituting another instantiation of the emulation and its interface, the validation software itself can be tested, with the help of both proper and selected improper variation controls applied to the "UUT" emulation.

In operation, the system executive functions as the user layer (a psuedo layer 7.5), along with the user interface and the test package data. An event queue functions as the command and data channel, in both directions, between the system executive

TABLE B.1

CCITT RECOMMENDATIONS BY ISO LAYER

FOR GROUP 4 FACSIMILE

APPLICATION LAYER NO CURRENT RECOMMENDATION		LAYER 7
PRESENTATION LAYER CCITT T.73		LAYER 6
SESSION LAYER CCITT T.62		LAYER 5
TRANSPORT LAYER CCITT T.70		LAYER 4
NETWORK LAYER CCITT X.25		LAYER 3
LINK LAYER CCITT X.25		LAYER 2
PHYSICAL X.24 PSTN	LAYER X.21 PSPDN	LAYER 1

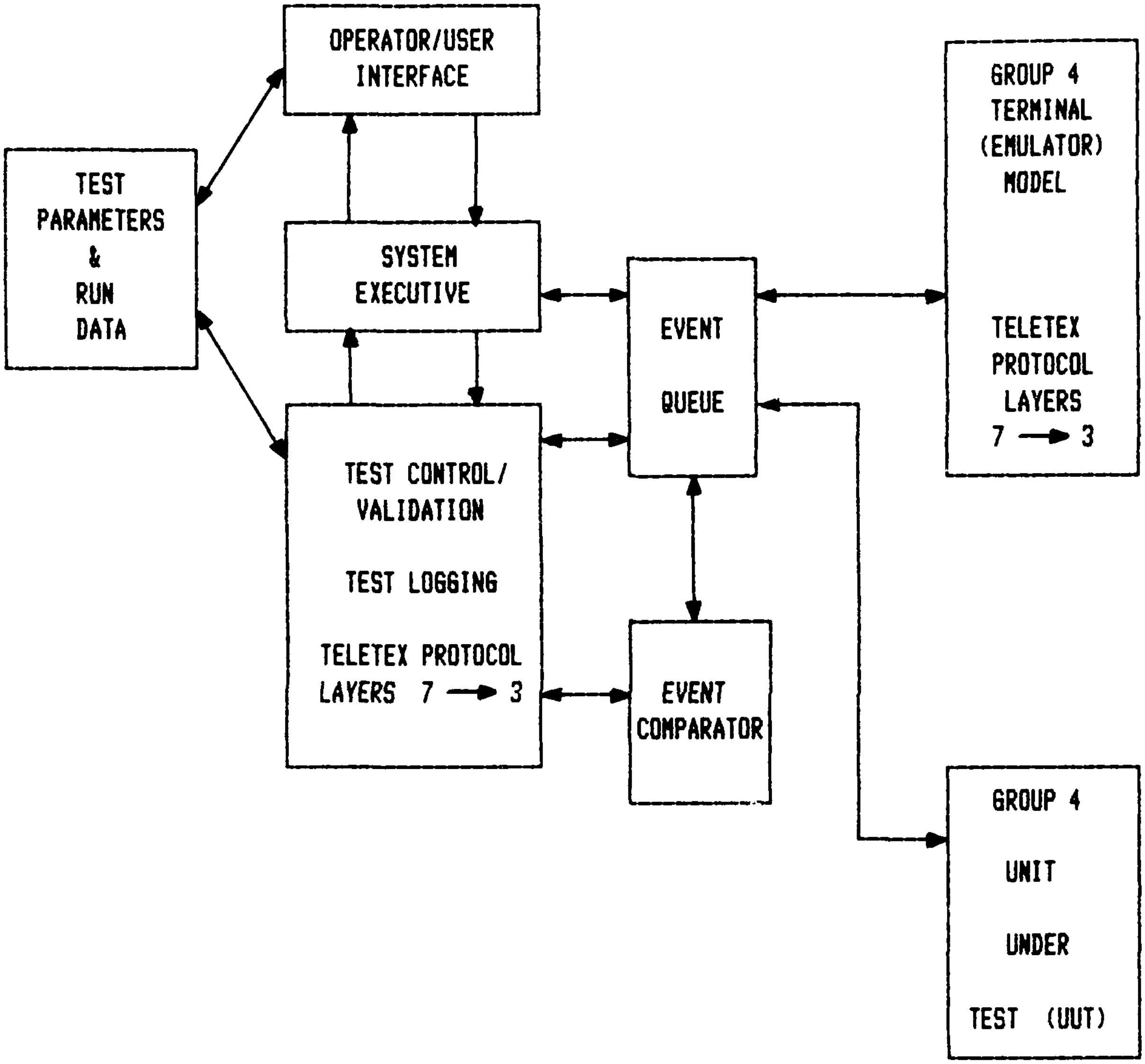

OPERATOR/USER INTERFACE
TEST PARAMETERS & RUN DATA
SYSTEM EXECUTIVE
EVENT QUEUE
GROUP 4 TERMINAL (EMULATOR) MODEL
TELETEX PROTOCOL LAYERS 7 → 3
TEST CONTROL/ VALIDATION
TEST LOGGING
TELETEX PROTOCOL LAYERS 7 → 3
EVENT COMPARATOR
GROUP 4 UNIT UNDER TEST (UUT)

FIGURE B.1 GROUP 4 FUNCTIONAL BLOCK DIAGRAM

and the two validation instantiations; the queue, in effect, functions as the link between the software portion of the system resident on the validation processor and the UUT.

The system executive starts and stops processes, in response to commands via the user interface and completion (successful or not) indications from the validations, and also examines the event queue to determine which modules to poll for action. Each module, when polled, modifies a state (or substate) or moves data as appropriate; the comparator is then called to determine if the action taken was permissible, via comparison with the "good" model, in some cases forcing the latter to match the actual UUT's (or in test mode, the possibly faulted validation's) action, if a permissible variation.

B.1.1.1 Design Philosophy

Thoroughness of testing, and the fineness of detail in the results obtained from tests, were the driving principles in the design of the system. This consideration, reinforced by structured programming principles, dictated that all actions which are significant (in protocol terms) be made visible by small modules. In order to make the system capable of maintenance and enhancement, modules were aligned to the layering (in the OSI sense) and to the CCITT protocol recommendations which they implement. In this way, the scope of the system as a whole can be easily broadened, and modifications in the CCITT protocol can be incorporated with relative ease.

Fortran 77 was chosen as source language for the validation system because of its portability, efficiency and its structuring (especially with MIL-STD-1753 enhancements). Its support for serially reusable modules permits, and redundancy/inconsistency considerations strongly advise, modules dedicated to protocol functions which are usable by both the UUT and the emulation subsytems, on both sides of the interface. Fortran's COMMON blocks are used to store states, substates, and overall status. The event queue is used not only as a "mailbox" between modules for interlevel commands, but also as a journal for the actions taken by the UUT and emulations of it. With the use of multiple linked lists, each corresponding to a specific event, maintained by well tried "heap" storage management techniques, each module quickly performs its intended function.

B.1.1.2 Design Approach

A top-down methodology was followed in the software design of the validation system. The system itself was structured as "open-ended", using small modules. Each module when coded was essentially complete, but those of its functions which were not needed in the present description of the system resulted in direct or indirect invocations of "stubs" or placeholder modules. When it is necessary to add a given function to the system, the modules completing that function can be substituted for the stubs.

The event queue approach was chosen to bring as many protocol actions as feasible out into the open, where the performance of the UUT can be compared in detail with a properly acting model supposedly compatible with it. It permits detailed error reports which not only back up invalidity decisions but also aid the agency requesting the validation, without the use of difficult-to-implement protocol conformance evaluations in the large. The approach also improved the open-endedness of the system, particularly with regard to added functionality and CCITT recommendation revisions (e.g. S.70/T.70,S.62/T.62).

B.1.2 Validation System Description

From a functional point of view, the validation system is comprised of the following parts:

- Operator (user) Interface
- Test Package
- System Executive
- UUT Subsystem and status/buffer stores
- Emulator Subsystem and status/buffer stores
- UUT/Emulator Comparator
- Event Queue and allied management software

The following details the part played in the operation by each of these subsystems.

B.1.2.1 Operator Interface

This set of modules provides the means by which specific tests can be selected and initiated, and the results of tests returned, in the form and in detail requested by the operator. It also provides the operator step-by-step instructions for normal UUT validations, including selection of alternate protocol where appropriate. For maintenance and diagnostic purposes, the operator may also choose between parallel and serial UUT and emulation operation, and may also compare a selectively faulted emulation with an unfaulted model. Internal to the system, this subsystem maintains a table of testing options, and calls the system executive to start, resume, or wrap up a test as instructed.

B.1.2.2 Test Package

While no modules are strictly part of the test package, this must be considered part of the system as a whole. This package will normally reside on auxiliary storage, and can be easily substituted for special purposes. Basically, it consists of test data for transmission, plus control information for selecting modes of operation on various levels. These modes include not only permissible alternatives but also invalid ones to test the UUT's capability to react properly to protocol errors.

B.1.2.3 System Executive

When invoked by the operator interface, the system executive initiates the selected task by obtaining testing information from the test package, storing it and initializing the event queue as appropriate. Thereafter, it polls the linked lists which make up the event queue and calls the appropriate modules to take action. On test completion or an abort condition signaled by the comparator subsystem, the general nature of the test result data can be printed or otherwise provided.

Executive polling can be done in two ways, roughly describable as parallel and serial. In the parallel mode, the

UUT and the emulation are kept essentially in step with one another; the UUT is blocked from getting more than a single significant protocol event ahead of the emulation. In this mode, the UUT does not proceed to step N+1 until the emulator has taken step N and its action compared with the corresponding UUT step. This mode is particularly useful for detailed examination of operation, especially in debugging.

The serial mode, on the other hand, allows UUT actions to have priority over emulation actions. letting the UUT "run free", so to speak. The emulator, and the comparison of its actions with the UUT, are handled as time is available, using the event queue as the UUT "history" medium. Serial mode may be required for some terminals and modes of operation; for emulation-to-emulation comparisons, the parallel mode is obviously preferable.

B.1.2.4 UUT Subsystem

This subsystem consists of a set of modules, shared with the emulator subsystem, which implement the protocols and functions called for at each layer of the OSI model. What actually dedicates it to the UUT (or an emulation of one) is the functional incorporation of the UUT and its hardware interface into the system, and the stored status, buffers, and linked lists specifically associated to the UUT or its standin. The procedure modules, as such, function as routines dedicated to their protocol implementation and transmission functions, rather than that of interfacing to the UUT or providing a control against which the UUT performance can be evaluated.

Each module performs one or more actions corresponding either to protocol-specified change of state or substate, or performs inter-layer translation functions. A module is thereby identified with specific sections or subsections of the protocol recommendation which it implements; its actions, rather than being "hard-coded", are driven by a decision table closely reflecting the "state diagrams" (when available) included with the CCITT recommendations, even to the state numbers and other annotations. The decision tables, fixed at compile time, are supplemented by parallel mask and vector tables which can modify actions either to reflect alternative actions or transmission paths, or to force improper actions to be taken, either to test the UUT's reaction to them, or for debugging purposes, especially in emulation vs emulation tests. These supplementary tables are modified during execution to implement alternate protocol choices, hardware vs software module implementations, and "error-force" option selections.

The heart of the UUT subsystem, as such, is the stored data which reflects the current states and substates of the UUT transmission in progress, the linked lists containing, by layer and direction, the history of the transmission, and the transmitted data itself. Substates, as far as the software is concerned, simply provide a more detailed description than the CCITT-defined states as such; the designation was chosen to keep the coarser states in line with the CCITT specifications. Relative to states, substates record such details as timeout counts, intermediate status in combine/divide operations, other data needed to define fully the status of a given transmission.

The linked lists provide the main mechanism by which the protocol modules communicate with one another. In operation, the executive polls all linked lists for unhandled events; when one is found, the module "handles" the event, usually marks it as "done" and places another event on another linked list for some other module to handle, modifying the stored status accordingly. In cases where the correspondence between "input" and "output events is not one-to-one, the module may delay "signing off" on the input event or placing another event on its own input linked list as is appropriate, to guarantee that it be polled to complete its function. In any case, the action taken by a protocol module on one invocation is scaled down to a maximum of one event in or out.

B.1.2.5 Emulator Subsystem

The emulator subsystem shares all its protocol procedure modules (except where replaced by hardware/firmware links) with the UUT subsystem; the difference is in the status and buffer stores dedicated to the subsystem, the linked lists which provide layer interfaces, and the method by which supervisory control over it is exercised. Each protocol module is ignorant whether it is performing its function for the UUT or emulator, but is provided with the status tables, linked lists, etc., peculiar to one or the other by the calling executive. The basic decision tables used are the same as for the UUT, but supplementary mask and vector tables may differ, according to the purpose of particular tests, and the double use of emulator modules on both the DTE and DCE sides of the transmission, for example when the effect of one sided protocol errors are being tested. The difference in supervisory control is implemented by the comparator subsystem, working with the executive, as described below.

B.1.2.6 UUT/Emulator Comparator

The job of the comparator subsystem is to keep the UUT and emulator systems in line with one another, comparing their actions on an event-by-event basis, and reporting on serious discrepancies. In order to do this, actions taken by protocol modules on both sides are "filtered" through a comparator module which takes differential actions, depending on the "side" from which the action emanates. On the UUT side, in serial mode actions are allowed to proceed; in parallel mode, an action may be held up until the emulator side has reacted to the corresponding stimulus. This is accomplished by the protocol module placing its generated events on the comparator module's stimulus list; the comparator will pass it on (by relinking) when and if appropriate.

Once both sides have reacted to corresponding stimuli, results are compared. If they match, the process is allowed to continue with no special action being taken. If the action taken by the UUT side is a valid alternative to that of the emulator, the latter's action is modified to match the UUT. Otherwise, an error report is generated and the entire process is aborted or modified and forced to continue as is appropriate to the seriousness of the error and pertinent operational modes as set by the operator.

In cases where only one side is a software module whose stimuli and responses are available to the comparator (for example, the DTE-end emulator paralleled to the UUT itself), no attempt is made to keep actions in line. Instead, the "small" actions on the "soft" side are assumed to match the other side, and alignment and comparison are deferred to a layer where both sides are available. No error reports are generated where mismatches can not be diagnosed of course; however, the comparator subsystem, through suitable interface modules, will merge error detections passed on by hardware/firmware modules into the same error report list used by software/software mismatch reports.

B.1.2.7 Event Queue and Associated Management Facilities

Although each event "belongs" to a specific side, layer, and module at any given time, its structure is common to the overall process itself. Storage for each event is allocated in space available to all modules, via common allocation/garbage collection routines. For auditing purposes, all events are linked by time of generation; for functional purposes, a second linkage is used to connect them with previous and successive events at the same and neighboring layers. This latter connection is modified (by relinking) by comparator modules in order to let a process continue or to "roll back" a given action.

Each event contains, at a minimum, the following information:

(1) An indication of the nature of the event, including codes for layers involved;

(2) An indication of the "side" (UUT vs emulation), "end" (DCE vs DTE), and direction of the event;

(3) Clock time for the event;

(4) List linkages (on time and functional bases);

(5) Linkages to data, where appropriate;

(6) Indications of state changes associated with the event.

Associated with the event queue are not only the allocation routines mentioned above, but also other service routines performing such chores as relinking lists and similar functions. A real time clock of adequate precision is used to provide event timing for timeout sequences and similar functions. Similarly, for reporting purposes, routines are required for editing compact error reports from the comparator subsystem into terms the operator can recognize.